

Universität Karlsruhe
Fakultät für Informatik
76128 Karlsruhe

Forschungs- und Arbeitsgebiete des Instituts für Telematik

April 1995

Professor Dr. Dr. h.c. Gerhard Krüger
Günter Schäfer (Hrsg.)

Universität Karlsruhe
Institut für Telematik

*Universität Karlsruhe
Institut für Telematik
Professor Dr. Dr. h.c. G. Krüger*

Klausurtagung

in

Oberuhldingen

vom 11. bis 13. April 1995

Kurzfassung

Der vorliegende Bericht gibt einen Überblick über aktuelle Forschungsarbeiten des *Instituts für Telematik der Universität Karlsruhe* in den Bereichen Hochleistungskommunikation, verteilte Systeme und Telekooperation. Er ist in zwei Teile gegliedert. Der erste beschreibt die persönlichen Interessensgebiete der wissenschaftlichen Mitarbeiter. Danach folgt eine Darstellung der Kooperationsprojekte des Instituts.

Im Forschungsbereich Hochleistungskommunikation liegen die Schwerpunkte bei fortgeschrittenen Kommunikationssystemen, innovativen Subsystemarchitekturen und auf Managementaspekten. Der Teilbereich fortgeschrittene Kommunikationssysteme befaßt sich mit Dienstmodellen, die Anforderungen neuartiger Anwendungen berücksichtigen (z.B. Multicast, Qualitätsparameter), der adäquaten Unterstützung dieser Dienste durch Protokollfunktionen sowie alternativen Implementierungsarchitekturen. Im Bereich innovative Subsystemarchitekturen steht die effiziente Realisierung von speziellen Kommunikationsfunktionen in Hochleistungsszenarien im Mittelpunkt (z.B. durch Integration von Hardwarekomponenten). Der Managementbereich umfaßt traditionelles Netzwerkmanagement und Monitoring, insbesondere in Hochleistungsnetzen (FDDI, ATM) sowie qualitätsorientierte Managementfunktionen.

Der Forschungsbereich „Verteilte Systeme“ deckt am Institut für Telematik das verteilte Datenmanagement, spezielle Systemunterstützung für verteilte Anwendungen und Agenten-modellierte Ansätze ab. Im ersten Bereich werden Konzepte objektorientierter Datenbanken unter den Aspekten der Adaptivität von Systemverhalten, der Integration mobiler Benutzer sowie der adaptiven Wissenspropagierung untersucht. Im zweiten Bereich stehen dedizierte Systemmechanismen wie lokationstransparente Aufrufe und dynamische Objektmigrationen in einer verteilten C++-Umgebung sowie Kooperationsmechanismen für gemeinsames Arbeiten im Rahmen verteilter Konferenzsysteme im Mittelpunkt. Im dritten Bereich wird als Erweiterung des objektorientierten Ansatzes der Einsatz von verteilten, kooperierenden Agenten in verteilten Systemen untersucht.

Der Bereich Telekooperation arbeitet mit seiner Forschung an einer integrierten Softwaretechnik für kooperative Benutzer. Schwerpunkte liegen auf dem Gebiet multimediale Konferenzsysteme, computergestützte Koordination von Benutzergruppen, multimodale Benutzungsschnittstellen und mobile Anwendungen; diese Forschung basiert dabei unmittelbar auf Arbeiten aus den Bereichen verteilte Systeme und Hochgeschwindigkeitskommunikation.

Abstract

This report gives an overview of current research topics at the *Institute of Telematics of the University of Karlsruhe*. It covers work in the areas of high performance networking, distributed systems and telecooperation, and comprises two parts. After presenting the research interests of the scientific staff in the first part, a brief description of the institutes cooperation projects is given.

In the area of high performance communication the main topics of interest are advanced communication systems, innovative subsystem architectures, and management aspects. On the side of advanced communication systems, one deals with service models that take into account the requirements of emerging applications (e.g., provision of multicast, parameters for quality of service), adequate support of these services by protocol functions, and alternative implementation architectures. In the area of innovative subsystem architectures, the efficient realization of specific communication functions in high performance scenarios (e.g., by integration of hardware components) is of interest. The work in the management area focuses on traditional network management as well as on monitoring, especially in high speed networks (FDDI, ATM), and on management functions related to quality.

The research work of the distributed systems group focusses on data management, system support for distributed applications and agent-based approaches. In the first part our main interests are concepts for building an object-oriented database system as an intelligent information backbone enabling adaptive system behaviour, integrating mobile users, and spreading the knowledge about available data services. Secondly, we investigate an extended distributed object-oriented environment supporting a uniform object model, location independent invocation and dynamic migration of fine-grained objects. Another aspect is a multimedia collaboration system, which allows individuals to participate in an audio and video conference. In the third part, a new effort concerns agent based distributed systems, in which autonomous agent objects cooperate by following their own flow of control.

The Telecooperation group works on an integrated software technology for cooperating humans. Focus areas are multi-media conferencing systems, computer supported coordination of multi-user groups, multi-modal user interfaces and mobile computing; this research is based on distributed computing systems and high-speed networking technology.

Vorwort

Der vorliegende Tagungsband basiert auf der fünften Klausurtagung des Instituts für Telematik vom 11. bis 13. April 1995 in Oberuhldingen am Bodensee. Er umfaßt sowohl Kurzbeschreibungen von den aktuellen Tätigkeitsgebieten der wissenschaftlichen Mitarbeiter am Institut, als auch eine Darstellung der Kooperationsprojekte, an denen das Institut beteiligt ist.

Die Kürze der einzelnen Beiträge erlaubt einen schnellen, aber prägnanten Einblick in die einzelnen Arbeitsgebiete. Der Preis dafür ist die Vernachlässigung konzeptioneller und technischer Detailinformationen, ohne die allerdings wissenschaftliche Arbeit nie praxisnahe Resultate liefern könnte. Daß auch diese Details existieren, zeigt ein Blick in die zitierten Veröffentlichungen, das Durcharbeiten der Studien- und Diplomarbeiten bzw. Dissertationen in den einzelnen Gebieten oder auch der persönliche Austausch mit den jeweiligen Autoren. Der vorliegende Band kann dabei eine zuverlässige Orientierungshilfe im weiten Feld der wissenschaftlichen Arbeit des Instituts sein.

Dieser Bericht verdankt seine Existenz der gemeinsamen Anstrengung aller Mitarbeiter des Instituts. Viele \LaTeX -Abstinenzler haben die Mühe auf sich genommen, ihre Dokumente zu konvertieren und dabei vor allem in die Anpassung der Bildformate viel Arbeit investiert. Ohne ihren Einsatz wäre das jetzige homogene Bild des Bandes nie zustande gekommen.

Karlsruhe, im April 1995

Günter Schäfer

Inhaltsverzeichnis

I Persönliche Beiträge

<i>Torsten Braun:</i>	
Protokollimplementierung mit Integrated Layer Processing	3
<i>Georg Carle:</i>	
Zuverlässige Gruppenkommunikationsdienste in ATM-Netzen	7
<i>Stefan Dresler:</i>	
ATM und IPng	11
<i>Oliver Frick:</i>	
Anwender-Kooperation in verteilten, multimedialen Umgebungen	15
<i>Hans-Werner Gellersen:</i>	
Softwaretechnik für flexible und multimodale interaktive Anwendungen	19
<i>Stefan Gessler:</i>	
Informationen in mobiler Umgebung	23
<i>Arnd G. Grosse:</i>	
Evolution in der Datenbanktechnologie	27
<i>Jörn Hartroth:</i>	
Kooperation autonomer Agenten	31
<i>Markus Hofmann:</i>	
Dienstgüteunterstützung für Gruppenkommunikation	35
<i>Ludwig Keller:</i>	
Zuverlässige Dienste in offenen Systemen: Anforderungen und Konzepte	39
<i>Dietmar Kottmann:</i>	
Abgekoppelte Objekte im Mobile Computing	43
<i>Günter Schäfer:</i>	
Einsatz des OSI-Management-Framework im Bereich Anwendungsmanagement	47
<i>Jochen Schiller:</i>	
CHIMPSY: Ein modulares Prozessorsystem für die Hochleistungskommunikation	51
<i>Claudia Schmidt:</i>	
Anwendungsorientierte Kommunikationsdienste	57
<i>Jochen Seitz:</i>	
Integration des Netzwerkmanagements in das Management verteilter Anwendungen	61
<i>Jörg Sievert:</i>	
Unterstützung von Kooperation in verteilten Systemen	65
<i>Burkhard Stiller:</i>	
Signaling in ATM Networks — Features and Protocols	69
<i>Volker Vogelmann:</i>	
Visualisierung vernetzter Systeme	75
<i>Hajo R. Wiltfang:</i>	
Moderne Hochleistungsnetze am Institut für Telematik	81
<i>Sonja Zwißler:</i>	
Kommerzielle Nutzung von Netzen	85

II Projekte

<i>Stefan Gessler:</i>	
Mobile Community & Workflow Initiative (MCW)	91
<i>Jörg Sievert:</i>	
Multimedia Collaboration	95
<i>Sonja Zwißler:</i>	
Mensch-Maschine Interface im multimedialen Verwaltungsbüro	97
<i>Stefan Gessler:</i>	
Der BERKOM Multimedia Multimedia Mail Teledienst (MMM)	101
<i>Markus Hofmann, Claudia Schmidt:</i>	
Das BerKom-II-Projekt MMT	105
<i>Stefan Gessler, Robert Wicke:</i>	
Multimediale Online-Dienste nach dem kombinierten WWW-Ansatz	109
<i>Arnd G. Grosse, Ludwig Keller, Dietmar A. Kottmann:</i>	
SFB 346: Rechnerintegrierte Produktion und Fertigung von Bauteilen	111
<i>Burkhard Stiller:</i>	
CogPiT — Configuration of Protocols in TIP	115

III Anhang

Literatur	123
Veröffentlichungen des Institutes	133

Persönliche Beiträge

Protokollimplementierung mit Integrated Layer Processing

Torsten Braun

Institut National de Recherche en Informatique et en Automatique (INRIA), Sophia Antipolis

1 Einführung

In den letzten Jahren stieg die Verarbeitungsleistung moderner RISC-Prozessoren in einem stärkeren Maße als die Zugriffsgeschwindigkeit auf den Hauptspeicher an. Insbesondere in Kommunikationsarchitekturen mit zahlreichen Datenmanipulationsfunktionen, d.h. Funktionen, die lesend oder schreibend auf Benutzerdaten zugreifen, wird der Zugriff auf den Hauptspeicher zum wesentlichen Engpaß. Um diesen Engpaß zu beseitigen, kann zunächst versucht werden, Hauptspeicherzugriffe durch Eliminierung von Kopieroperationen zu beseitigen. Es ist offensichtlich, daß dieser Ansatz nur begrenzt zum Erfolg führen kann. Eine weitere Möglichkeit besteht darin, parallele Implementierungsarchitekturen zu verwenden, und beispielsweise für die Verarbeitung von Benutzer- und Kontrolldaten sowie für den Sende- und Empfangsteil separate Verarbeitungseinheiten einzusetzen. [BS94, Bra94a, BSSZ94, Bra95a, Bra94b, BSZ94b, BSZ94a]. Ein weiterer möglicher Ansatz besteht in der Verwendung sogenannter Cache-Speicher. Cache-Speicher sind um ein Vielfaches schneller als der Hauptspeicher, in der Regel ist der Cache-Zugriff kaum langsamer als auf die Prozessor-Register. Dadurch vermeiden sie Wartezyklen, die in der Regel beim Zugriff auf den Hauptspeicher erforderlich sind. Die Cache-Speicher sind entweder direkt auf der CPU (on-chip/first-level Cache) oder extern (off-chip/second-level Cache) untergebracht. Nachteil bei den Cache-Speichern sind derzeit die deutlich höheren Kosten und ihre begrenzte Größe. Beispielsweise beträgt die Größe des first-level Cache bei einem Sun SuperSPARC Prozessor 16 kbyte für Daten und 20 kbyte für Instruktionen, beim DEC Alpha 21064 Prozessor sind beide first-level Caches lediglich 8 kbyte groß. Aufgrund der begrenzten Größe ist es deshalb entscheidend, daß immer nur die benötigten Daten im Cache-Speicher vorhanden sind. Insbesondere, wenn verschiedene Teile eines Kommunikationssystems (z.B. Ebenen) in unterschiedlichen Prozessen oder Prozeduren implementiert sind, ist die Wahrscheinlichkeit, daß die benötigten Daten während der gesamten Verarbeitung im Kommunikationssystem immer im Cache liegen, geringer als bei einer integrierten Implementierung, bei der gegebenenfalls mehrere Ebenen in einer Prozedur zusammengefaßt werden.

2 Integrated Layer Processing und Application Layer Framing

Integrated Layer Processing (ILP) ist ein Implementierungskonzept, welches dem Implementierer von Kommunikationsprotokollen erlaubt, verschiedene Datenmanipulationen in möglichst wenigen, im besten Fall einer oder zwei Verarbeitungsschleifen (ILP loop) zusammenzufassen. Dadurch sollen zum einen unnötige Datenkopieroperationen eliminiert werden, und zum anderen soll dadurch die Wahrscheinlichkeit erhöht werden, daß sich die Daten während der Verarbeitung durch die einzelnen Datenmanipulationsfunktionen des Kommunikationssystems immer im Cache befinden.

Idealerweise werden bei einer ILP loop die Daten einmal vom Hauptspeicher gelesen, die erforderlichen Datenmanipulationsfunktionen sämtlicher Ebenen werden auf der gelesenen Einheit (in der Regel wenige Bytes, z.B. 4/8 Bytes) ausgeführt und das Ergebnis wird schließlich in den Zielspeicher geschrieben. Sämtliche Datenmanipulationsfunktionen arbeiten dabei auf Registern oder dem Cache-Speicher.

Im Gegensatz dazu wird bei einer nicht-integrierten Implementierung bei jeder Datenmanipulationsfunktion das gesamte Paket gelesen, bearbeitet und schließlich wieder in den Speicher zurückgeschrieben. Gegebenenfalls werden die Pakete zur Übergabe zwischen den Funktionen von einem Speicher in einen anderen kopiert. Die Kopiervorgänge lassen sich aber in der Regel eliminieren, wenn die verschiedenen Funktionen im gleichen Adreßraum implementiert sind. Außerdem wurde in Experimenten festgestellt, daß sich auch bei einer nicht-integrierten Implementierung zumindest für nicht zu große Pakete ein zufriedenstellendes Cache-Verhalten erreichen läßt, sofern man entsprechend dem Application Layer Framing (ALF) Konzept die Paketgrenzen bei der Protokollverarbeitung nicht durch Multiplex- oder Segmentierungsfunktionen verändert [BD95]. ALF bedeutet im wesentlichen, daß die Anwendung die Benutzerdaten in für sie passende Einheiten (application data units, ADUs) unterteilt. Diese ADUs werden im gesamten Kommunikationssystem nicht weiter zerteilt und werden nicht mit anderen Einheiten wieder kombiniert. Dadurch werden auch aufwendige Pufferoperationen in Warteschlangen bei der Übergabe

der Daten zwischen den Ebenen vermieden. Nicht zuletzt deshalb ist die Wahrscheinlichkeit, daß die Daten im Cache-Speicher beileiben, relativ groß. Vorausgesetzt, daß die Größe der Cache-Speicher zukünftig wieder zunimmt, erscheint ALF auch ohne ILP ein interessantes Konzept zu sein.

In einem einfachen Beispiel wurde durch die Anwendung des ILP-Konzepts bei der Integration von Marshalling und Prüfsummenberechnung eine Leistungssteigerung von 40 Prozent erreicht.

ALF wurde zusammen mit ILP von Clark und Tennenhouse entwickelt [CT90b]. Auf diesen beiden Konzepten basierende Kommunikationssysteme werden in dem europäisch australischen Forschungsprojekt HIPPARCH entwickelt. An diesem Projekt sind neben INRIA auch SICS, University College London und die University of Technology Sydney beteiligt [CH94].

3 Probleme und Nachteile von ILP

ILP kann nur unter bestimmten Bedingungen und Voraussetzungen sinnvoll eingesetzt werden. In einer ILP loop dürfen keine Operationen vorkommen, die PDUs zerteilen oder kombinieren, d.h. das ALF-Konzept ist eine grundlegende Bedingung für ILP. Allerdings ist das Vermeiden von Multiplexen nicht nur für ILP sondern auch für andere effizienzsteigernde Implementierungstechniken und für die Einhaltung von verbindungspezifischen Dienstgten eine Notwendigkeit.

Eine andere Beschränkung ist, daß in einer ILP loop nur sogenannte 'ordering constrained' Protokollfunktionen angewendet werden können. Protokollfunktionen dieser Art erfordern, daß die Verarbeitung einer Protokollfunktion seriell vom Beginn zum Ende eines Pakets ablaufen muß. Beispiele solcher Funktionen sind CRC-Prüfsummen oder Stream-Cipher-Verschlüsselungsalgorithmen. Die TCP-Prüfsumme oder Block-Cipher-Verschlüsselungsalgorithmen sind Gegenbeispiele, d.h. sie sind gut für ILP geeignet. Es ist aber auch bei den nicht 'ordering constrained' Protokollfunktionen notwendig, daß die Größe der Header a priori bekannt ist, da die Datenmanipulationsoperationen auf bestimmte Grenzen (z.B. 8 Bytes für Verschlüsselungsalgorithmen oder 2 Bytes für die TCP-Prüfsumme) ausgerichtet sein müssen.

Ein weiteres Problem besteht darin, daß Datenmanipulationsfunktionen den Header-Inhalt beeinflussen. Beispielsweise wird das Ergebnis der TCP-Prüfsummenberechnung in den TCP-Header geschrieben. Längenfelder, welche die Länge der Benutzerdaten anzeigen, sind weitere Beispiele. Die Abhängigkeiten zwischen Daten und Header verhindern, daß zunächst der Header vollständig verarbeitet und danach die Benutzerdaten. Ansätze zur Lösung dieses Problems werden in [BD95, AL93] vorgeschlagen.

Weitere ILP-Nachteile bestehen in der begrenzten Möglichkeit, eine ILP-Implementierung flexibel zu gestalten. Eine effiziente ILP-Implementierung erfordert die Benutzung von Macro-Inlining an Stelle von Funktionsaufrufen. Funktionsaufrufe würden den Performance-Gewinn durch ILP wieder aufheben. Dies bedeutet, daß die Art der verwendeten Protokollfunktionen zur Übersetzungszeit bekannt sein muß und somit eine dynamische Rekonfiguration des Kommunikationssystems aus Leistungsgründen nur in einem unbefriedigendem Maß möglich ist.

Weitere Probleme bestehen in Modularitätsaspekten, der unterschiedlichen Verarbeitungseinheiten der verschiedenen Datenmanipulationsfunktionen und den Abhängigkeiten zwischen Kontroll- und Datenteil innerhalb eines Kommunikationssystems. Lösungsansätze werden in [BD95, Bra95b, AL93] diskutiert.

4 ILP-Implementierungsexperiment

Um den erzielbaren Nutzen durch ILP abschätzen zu können, wurde ein umfangreicheres Experiment in einer realen Workstation-Umgebung durchgeführt.

4.1 Ein ILP-Kommunikationssystem basierend auf TCP/IP

Als Beispielapplikation wurde ein einfacher Dateitransfer mit Verschlüsselungsfunktion über einer TCP-Implementierung im User Space implementiert. TCP wurde als linkbare Bibliothek realisiert, d.h. eine Anwendung (z.B. Audio- oder Videoanwendungen [BZ94]) muß die TCP-Bibliothek dazubinden [BDH95].

Beim Senden werden die Benutzerdaten nach dem Marshalling verschlüsselt und in den Übertragungswiederholungspuffer von TCP kopiert. Danach wird die TCP Prüfsumme berechnet und schließlich werden die Daten in den Kern-Puffer übertragen und von IP weiterverarbeitet.

Empfangene IP-Pakete werden von einem TCP-Kern-Teil an den korrekten Anwendungsprozeß weitergeleitet. Der TCP-Kern-Teil führt damit im wesentlichen Demultiplex-Funktionen aus und ist mit

UDP vergleichbar. Die Unterschiede bestehen im Fehlen eines speziellen Headers und in der Prüfsummenberechnung. Die Daten des TCP-Paketes wird direkt in den von der Anwendung gewünschten Speicher geschrieben, der Header wird in einen TCP-internen Puffer kopiert. Dieses Vorgehen ist nur dann möglich, wenn das TCP-Paket die Bedingungen des Header Prediction Tests erfüllt, d.h. z.B. der Header die erwartete Länge (keine Optionen) besitzt und die Sequenznummer mit der erwarteten übereinstimmt. Danach wird die Prüfsumme getestet, die Daten entschlüsselt und die Nachricht über eine Unmarshalling-Routine an die Anwendung übergeben.

Zur Erzeugung der Marshalling/Unmarshalling-Routinen wurde der MAVROS-Stub-Compiler verwendet [Hui90], der aus einer ASN.1-Beschreibung der Datentypen Routinen zur Transformation in das XDR Format erzeugt. Als Verschlüsselungsalgorithmus diente eine vereinfachte Version des SAFERK-64 Algorithmus [Mas94]. Dieser Verschlüsselungsalgorithmus soll besonders leistungsfähige Software-Implementierungen erlauben.

Zur Erzeugung der ILP-Implementierung wurde auf der Sendeseite Marshalling, Verschlüsselung, Kopieroperation und Prüfsummenberechnung integriert, indem die Marshalling-Routinen um entsprechende Macro-Anweisungen erweitert wurden. D.h. die aufgerufene Marshalling-Routine schreibt die Daten bereits verschlüsselt in den TCP-Übertragungswiederholungspuffer. Beim Empfangen wurde die Prüfsummenberechnung, Entschlüsselung und Unmarshalling kombiniert.

Eine automatische Erzeugung von ILP-Marshalling-Routinen wurde für den INRIA Stub Compiler (ISC) entwickelt. ISC verwendet zur Beschreibung der Datentypen die Programmiersprache C mit Annotationen, was insbesondere die Erstellung verteilter Anwendungen erleichtert.

4.2 Leistungsbewertung

Die ILP-Implementierung wurde in Leistungsmessungen mit den nicht-integrierten Versionen mit TCP im User Space und im Kernel Space verglichen. Die Experimente wurden auf verschiedenen SUN SP-ARCstations und DEC-AXP Workstations durchgeführt. Außerdem wurde mit verschiedenen Simulationswerkzeugen das Cache-Verhalten der Implementierung analysiert.

Leistungsmessungen Die Meßergebnisse zeigten bei der Dateitransfer-Applikation Leistungsvorteile von ILP im Bereich von 20 Prozent bei den Paketverarbeitungszeiten, d.h. der Zeit, welche die Verarbeitung eines Pakets zwischen Anwendung und Kern innerhalb des implementierten Kommunikationssubsystems erforderlich ist. Aufgrund des System-Overheads reduzierte sich der Leistungsgewinn beim Durchsatz auf ca. 10 Prozent. Allerdings wurde mit der nicht-integrierten Version mit TCP im Kern ein etwas höherer Durchsatz erzielt, obwohl auch hier die Paketverarbeitungszeiten höher waren als bei der ILP-Implementierung. Der Grund für den besseren Kern-Durchsatz ist in der niedrigeren Anzahl von Prozeßwechseln zu suchen, da bei der User-Level-Implementierung für jedes TCP-Paket (auch für reine Quittungen) ein Prozeßwechsel stattfindet. Vorschläge zur Umgehung dieses Problems sind in [Bra95b] zu finden.

Einen wesentlichen Einfluß auf die Ergebnisse hat auch die verwendete Rechnerplattform. So machte sich abhängig vom Betriebssystem der Nutzen von ILP sehr unterschiedlich bemerkbar. Bei effizienten Betriebssystemen wie Solaris war der System-Overhead wesentlich geringer als beispielsweise auf einer DEC-AXP Workstation mit OSF/1, was natürlich den Nutzen von ILP deutlicher erkennen ließ. Mit fortgeschrittenen Netzwerkadaptern (z.B. single-copy-Adapter) und deren effizienten Integration in die Systemumgebung werden die ILP-Vorteile noch deutlicher sichtbar werden.

In einem weiteren Experiment wurde der Verschlüsselungsalgorithmus durch einen sehr einfachen Algorithmus ersetzt. Dadurch stieg die Verbesserung des Durchsatzes durch ILP auf 15 Prozent bei einer Verbesserung der Paketverarbeitungszeiten um 35 Prozent. D.h. die Komplexität der Datenmanipulationsfunktionen beeinflußt den erzielbaren Gewinn spürbar. Bei komplexen Datenmanipulationen dominiert zwangsläufig die CPU-Geschwindigkeit und der langsame Speicherzugriff macht sich weniger stark bemerkbar. Hier scheint auf den ersten Blick ILP nicht allzu große Vorteile bringen, bei einer weiteren stärkeren Zunahme der CPU-Geschwindigkeit im Vergleich zur Speicherzugriffsgeschwindigkeit wird ILP jedoch zunehmend interessanter.

Simulation des Cache-Verhaltens Das Speicherzugriffs- und Cache-Verhalten der vermessenen Programme wurde mit verschiedenen Simulationswerkzeugen untersucht. Es zeigte sich dabei, daß die erzielten Leistungssteigerungen durch ILP eher auf der reduzierten Anzahl von Speicherzugriffen als auf einer Erhöhung der Cache-Hit-Rate basieren. Die eingesparten Speicherzugriffe entsprachen beim Senden und Empfangen von Daten 3-4 Kopieroperationen. In den meisten Fällen ging durch ILP zwar

auch die absolute Anzahl von Cache misses zurück, doch der relative Anteil der Cache misses war bei der nicht-integrierten Implementierung höher. Dieser zunächst überraschende Effekt hat zwei Ursachen: Zum einen lassen sich bestimmte Cache misses, wie das erstmalige Lesen und das letztmalige Schreiben in den Zielspeicher nicht vermeiden. Dies Operationen führen sehr häufig zu Cache misses. Der relative Anteil dieser Operationen an der Gesamtzahl der Speicherzugriffe erhöht sich bei der ILP-Implementierung. Der zweite Grund liegt in der Art der ausgewählten Datenmanipulationsfunktionen, im speziellen Beispiel am Verschlüsselungsalgorithmus. Der Algorithmus benötigt mehrere Tabellen zur Verschlüsselung, greift also nicht nur auf die Benutzerdaten während der Manipulationsoperationen zu. Bei der ILP-Implementierung wird die Verschlüsselung immer wieder durch andere Manipulationen unterbrochen und damit die Wahrscheinlichkeit, daß sich die benötigten Daten im Cache bleiben, reduziert. Es muß allerdings betont werden, daß die nicht-integrierte Implementierung derart gestaltet wurde, daß die Datenmanipulationsfunktionen dicht aufeinanderfolgen. Eine hohe Cache-Hit-Rate wird nicht zuletzt dadurch erreicht, daß auch die nicht-integrierte Implementierung nach dem ALF-Konzept realisiert wurde. Die Ergebnisse auf der DEC-AXP-Workstation bezüglich des Daten-Cache waren den SUN-Ergebnissen sehr ähnlich. Ein weiterer Nachteil war bei der ILP-Implementierung auf DEC-AXP zu sehen. Durch den extrem kleinen Instruktions-Cache verursachte die ILP-Implementierung deutlich mehr Cache misses. Eine detaillierte Darstellung der Ergebnisse findet sich in [BD95].

5 Diskussion und Ausblick

Das durchgeführte Experiment hat deutlich die Grenzen von ILP aufgezeigt, die das Konzept beim Einsatz in herkömmlichen Protokollen und in üblichen Implementierungsumgebungen mit sich bringt. Zwar wird ein gewisser Leistungszuwachs durch ILP erreicht, aber aufgrund der damit verbundenen Nachteile (nur gewisse Protokollfunktionen erlauben ILP, begrenzte Flexibilität, aufwendigere Implementierung) erscheint es fraglich, ob sich eine ILP-Implementierung lohnt. Die bloße Anwendung des ALF-Konzepts erscheint zunächst eine gute Alternative zu sein. Speziell bei leistungsfähigeren Workstations mit superschnellen CPUs und single-copy-Netzwerkadaptern erscheint ILP jedoch interessanter zu werden. Auch können verbesserte Protokollarchitekturen durch feste Header-Formate, auf die Basiseinheiten der Datenmanipulationsfunktionen abgestimmte Header-Formate, Demultiplex-Unterstützung durch eindeutige Verbindungsidentifikatoren sowie getrennte Kontroll- und Datenpakete ILP-Implementierungen wesentlich vereinfachen.

Auch erscheint ILP für Kommunikationssysteme mit aufwendigen Datenmanipulationen interessant, falls die teuren Datenmanipulationen durch programmierbare Hardware verarbeitet werden können. Oft werden nämlich für aufwendige Datenmanipulationsfunktionen Spezial-Hardware-Lösungen eingesetzt. Falls in einem Kommunikationssystem mehrere solcher Spezial-Hardware-Bausteine genutzt werden, sind mehrere Datentransfers vom Speicher zu diesen Spezial-Hardware-Bausteinen und zurück erforderlich. Wie in [Bra95b] vorgeschlagen, könnten die aufwendigen Datenmanipulationsfunktionen zu einer ILP loop zusammengefaßt und durch Anwendung des ILP-Ansatzes auf einem einzigen programmierbaren Baustein implementiert werden. Es bietet sich hier zur automatisierten Implementierung an etwas einen ILP-Stub-Compiler zu realisieren, der nicht wie die ISC-Erweiterung C-Funktionen sondern etwa VHDL-Funktionen generiert.

Zuverlässige Gruppenkommunikationsdienste in ATM-Netzen

Georg Carle

1 Motivation

Mit der Einführung einer neuen Netzgeneration, die auf dem Übermittlungsprinzip des Asynchronen TransferModus (ATM) basiert, findet in der Telekommunikation in mehrerer Hinsicht ein Paradigmenwechsel statt. Konventionelle Datennetze sind geprägt durch eine verbindungslose Vermittlung von Paketen mit variabler Länge sowie durch Kommunikationsprotokolle, die für niedrige Übertragungsraten und für hohe Fehlerraten entworfen wurden. Im Gegensatz dazu basieren ATM-Netze auf verbindungsorientierter Vermittlung von Zellen einer konstanten Länge von 53 Bytes und ermöglichen hohe Übertragungsraten über weite Entfernungen.

Auch auf Seiten der Anwendung läßt sich ein deutlicher Wandel beobachten. Während in konventionellen Anwendungen Punkt-zu-Punkt-Kommunikation überwiegt, basieren viele fortgeschrittene Anwendungen auf Kommunikationsformen, welche eine Gruppe von Kommunikationsteilnehmern involviert. Beispiele finden sich in den Bereichen verteilte rechnergestützte Gruppenarbeit, Konferenzsysteme, Verteildienste und verteiltes Rechnen. Diese Anwendungen fordern vom Netz unter anderem Dienste für die Kommunikation eines Senders mit mehreren Empfängern, was als Multicasting bezeichnet wird, sowie Dienste für mehrere Sender in einer Gruppe (Multipeer-Kommunikation). Häufig benötigen diese Anwendungen einen zuverlässigen Gruppenkommunikationsdienst, der Protokolle mit Fehlerkontrollmechanismen erforderlich macht. Existierende Protokolle für zuverlässige Gruppenkommunikation wurden im Hinblick auf konventionelle Netze entworfen und sind für den Einsatz in ATM-Netzen wenig geeignet. Ziel der vorliegenden Arbeit ist es, die beim Einsatz von ATM-Netzen geltenden neuen Randbedingungen sowie die neuen Anwendungsanforderungen der Gruppenkommunikation zu analysieren und darauf aufbauend neue Protokollmechanismen und neue ATM-Dienstkomponenten für die effiziente Erbringung zuverlässiger Gruppenkommunikationsdienste zu entwickeln.

2 Mechanismen für zuverlässige Gruppenkommunikation

Die zu entwickelnden Protokollmechanismen erbringen anwendungsspezifische Dienste und lassen sich somit gemäß der geschichteten Protokollarchitektur von ATM-Netzen der sogenannten ATM-Adaptionsschicht (ATM Adaptation Layer, AAL) zuordnen. Die Protokollmechanismen dieser Schicht operieren auf einzelnen Zellen sowie auf aus mehreren Zellen zusammengesetzten Rahmen. In dieser Arbeit werden erstmals multicastfähige Fehlerkontrollmechanismen für die Adaptionsschicht vorgestellt, während sich bisherige Arbeiten dieser Schicht auf zuverlässige Punkt-zu-Punkt-Kommunikation beschränkten. Vorhandene Arbeiten zu Protokollen dieser Schicht konzentrierten sich außerdem auf eine sehr geringe Zellverlustwahrscheinlichkeit. Wenn allerdings für Mehrpunktverbindungen mit einer sehr stoßartigen Verkehrscharakteristik eine geringe Zellverlustwahrscheinlichkeit garantiert werden soll, so müßten Netzressourcen in deutlich größerem Umfang reserviert werden, als im Mittel auch genutzt werden.

Gruppenkommunikationsdienste können in vielen Fällen somit nur dann effizient erbracht werden, wenn eine relativ hohe Zellverlustwahrscheinlichkeit in Kauf genommen wird. In diesen Fällen sind zur Behebung der Zellenverluste Kommunikationssysteme mit besonders effizienten Fehlerkontrollmechanismen dringend erforderlich. Heutige Realisierungen von Fehlerkontrollmechanismen führen allerdings häufig schon für den einfacheren Fall zuverlässiger Punkt-zu-Punkt-Kommunikation zu einem Leistungsengpaß. Bei zuverlässiger Gruppenkommunikation wird dieses Problem besonders deutlich, da sowohl der Verarbeitungsaufwand zur Protokollbearbeitung als auch die Fehlerhäufigkeit mit wachsender Gruppengröße zunimmt.

Als wesentliche Einflußgrößen für den Leistungsengpaß wurden die Parameter Empfängerzahl, Übertragungskapazität, Entfernung, Verlustwahrscheinlichkeit und Sendecharakteristik identifiziert [Car94a]. Sie können über mehrere Größenordnungen schwanken. Die Entwicklung von Protokollmechanismen mit einer guten Skalierbarkeit ist daher eine große Herausforderung. Wegen der stark unterschiedlichen Dienstanforderungen sowie der über einen großen Bereich schwankenden Randbedingungen bietet kein einzelner Protokollmechanismus für alle Fälle zufriedenstellende Leistungen, weshalb konfigurierbare und parametrisierbare ATM-Dienstkomponenten erforderlich sind [Car94b].

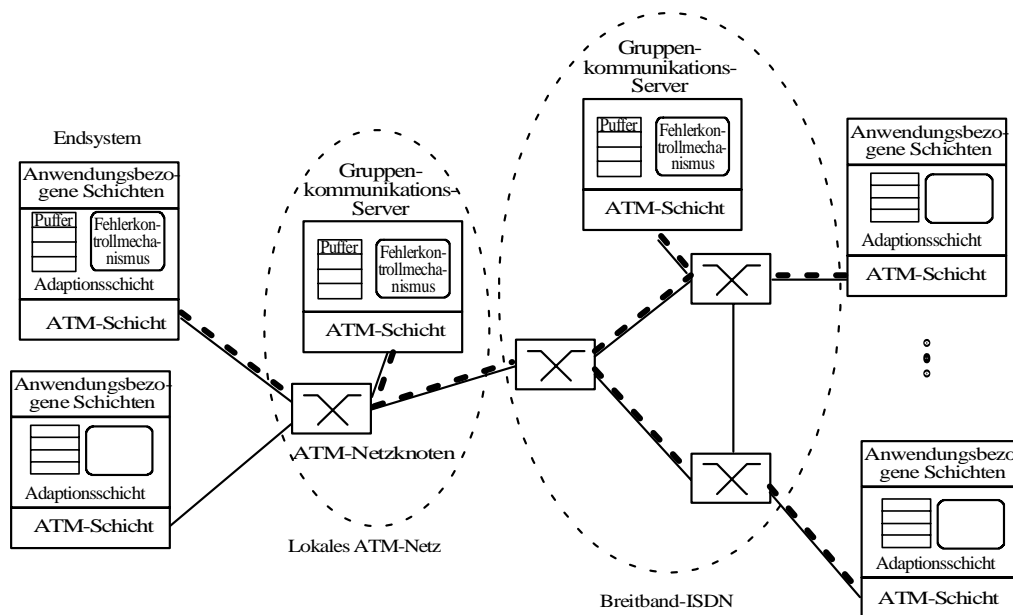


Abbildung 1. Adaptionsschicht und Gruppenkommunikationsserver mit Mechanismen zur Unterstützung zuverlässiger Multicast-Kommunikation

3 Erweiterte Adaptionsschicht und Gruppenkommunikationsserver

In dieser Arbeit wird das neuartige Konzept einer flexiblen, erweiterten Adaptionsschicht (Extended Adaptation Layer, XAAL) vorgestellt, die durch das Vorhandensein verschiedener Fehlerkontrollmechanismen für eine Vielzahl unterschiedlicher Szenarien die effiziente Erbringung zuverlässiger Gruppenkommunikationsdienste ermöglicht. Bisherige Konzepte sehen beim Auftreten von Zellverlusten üblicherweise eine Übertragungswiederholung des ganzen Rahmens vor. In dieser Arbeit werden leistungsfähigere Protokollmechanismen für zuverlässige Gruppenkommunikation entwickelt. Im Mittelpunkt steht eine systematische Untersuchung der Fehlerbehandlung auf Basis von Zellen bzw. Rahmen, der Nutzung von Redundanz für Vorwärtsfehlerkorrektur (Forward Error Correction, FEC) sowie der Verarbeitung von Quittungen.

Während in bisherigen Untersuchungen eine Fehlerkontrolle in den Endsystemen im Mittelpunkt stand, präsentiert diese Arbeit das Konzept eines Gruppenkommunikationsservers, mit dem die Fehlerkontrolle auch innerhalb des Netzes durchgeführt werden kann (Abb. 1). Dadurch läßt sich die Netzauslastung verbessern, ein Leistungsengpaß bei der Protokollverarbeitung vermeiden und eine geringere Auslieferungsverzögerung im Fehlerfall gewährleisten. Mit Mechanismen zum Multiplexen von Rahmen verschiedener Sender stellt der Server auch für die Multipeer-Kommunikation eine gute Skalierbarkeit sicher.

4 Implementierung und Leistungsuntersuchung

Quittungsverwaltung. Die Untersuchung von Mechanismen zur Quittierung ergab, daß eine Quittungsverwaltung auf Basis verketteter Listen mit wachsender Gruppengröße einen potentiellen Leistungsengpaß darstellt, für geringe Zellverlustwahrscheinlichkeiten aber einen sehr geringen Speicheraufwand besitzt. Eine Implementierung der listenbasierten Quittungsverwaltung in Software zeigt, daß diese Lösung nur für wenige Empfänger und niedrige Fehlerwahrscheinlichkeiten geeignet ist. Mit dem Entwurf eines speziellen Hardware-Koprozessors wird nachgewiesen, daß sich dieser Engpaß deutlich verringern läßt [CSS94]. Der Einsatz einer zusätzlichen Hardwarekomponente für die Quittungsverarbeitung ist allerdings wegen des erhöhten Realisierungsaufwandes in vielen Fällen unerwünscht.

Unter Vermeidung des teilweise hohen Verarbeitungsaufwands listenbasierter Quittungsverwaltung wurden für unterschiedliche Einsatzbereiche zwei neue Protokollmechanismen entwickelt. Beide zeichnen sich durch die Verwendung von Binärfeldern für die Quittungsverarbeitung aus, was auch für große Gruppen zu einem geringen Verarbeitungsaufwand führt. Die entwickelten Protokollmechanismen lassen sich in die XAAL und den Gruppenkommunikationsserver einbetten. Sie können anforderungsgesteuert selektiert und parametrisiert werden, woraus sich die geforderte Flexibilität der Gruppenkommunikationsdienste für eine Vielzahl von Anwendungen ergibt.

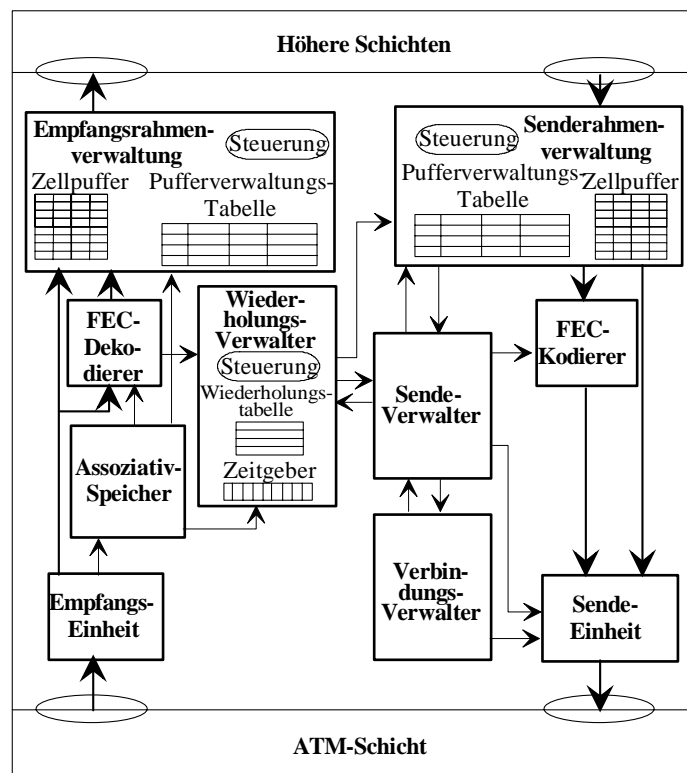


Abbildung 2. Funktionale Architektur der Adaptionsschicht für zuverlässige Multicast-Kommunikation RMC-AAL

Rahmenbasierte Übertragungswiederholung. Der erste Protokollmechanismus (Reliable Multicast Service Specific Convergence Sublayer, RMC-SSCS [Car94d]) läßt sich mit dem bereits standardisierten Protokollmechanismus zur Fehlererkennung des Adaptionsschichttyps AAL5 kombinieren. Er verwendet eine Übertragungswiederholung auf Rahmenbasis mit veränderlicher Rahmengröße. Damit kann für Mehrpunktverbindungen in Bereichen niedriger Zellverlustrate ein besonders geringer Bandbreitenzusatzaufwand erzielt werden.

Zellenbasierte Übertragungswiederholung und Vorwärtsfehlerkorrektur. Der zweite Protokollmechanismus (Reliable Multicast ATM Adaptation Layer, RMC-AAL [Car94c]) wurde für Bereiche höherer Zellverlustwahrscheinlichkeit sowie für Anwendungen, die besonders niedrige Verzögerungszeiten fordern, entworfen. Als erster Protokollmechanismus für die Adaptionsschicht ermöglicht er die gleichzeitige Verwendung von Übertragungswiederholung und Vorwärtsfehlerkorrektur. Das neuentwickelte FEC-Verfahren besitzt eine hohe Robustheit gegenüber bündelartigen Zellenverlusten. Es zeichnet sich außerdem durch einen geringen Verarbeitungsaufwand aus, so daß leistungsfähige Softwareimplementierungen möglich sind. Zur Erzielung einer besonders hohen Leistungsfähigkeit wurde für den FEC-Mechanismus ein spezieller Hardwarebaustein spezifiziert und mit Hilfe eines Hardwaresynthesewerkzeugs bis auf Gatterebene entworfen. Abb. 2 zeigt eine funktionale Architektur zur Implementierung von RMC-AAL.

Das FEC-Verfahren von RMC-AAL läßt sich mit einer rahmenbasierten Übertragungswiederholung kombinieren (ARQ rahmenbasiert mit FEC), was zu sehr geringen Verarbeitungskosten führt. Das FEC-Verfahren kann auch mit einem speziellen, zellenbasierten Übertragungswiederholungsverfahren (ARQ zellenbasiert mit FEC) kombiniert werden. Die dabei verwendeten Rahmenfragmente erlauben es erstmals, Zellenverluste auch für Verbindungen mit großer Pfadkapazität und stark schwankender Verbindungsqualität effizient zu beheben. Als dritten Operationsmodus erlaubt RMC-AAL, das zellenbasierte ARQ-Verfahren ohne FEC einzusetzen (ARQ zellenbasiert), womit sich der Bandbreitenzusatzaufwand von FEC vermeiden läßt. Für jeden dieser drei Betriebsmodi gibt es Einsatzbereiche, für die sich Vorteile bezüglich nutzbarer Bandbreite, mittlerer Verzögerung oder der Verarbeitungskosten erzielen lassen.

Leistungsbewertung. Extensive Analysen und Simulationen konnten für eine Vielzahl von Szenarien die hohe Leistungsfähigkeit der neuentwickelten Adaptionsschichttypen und des Gruppenkommunikations-

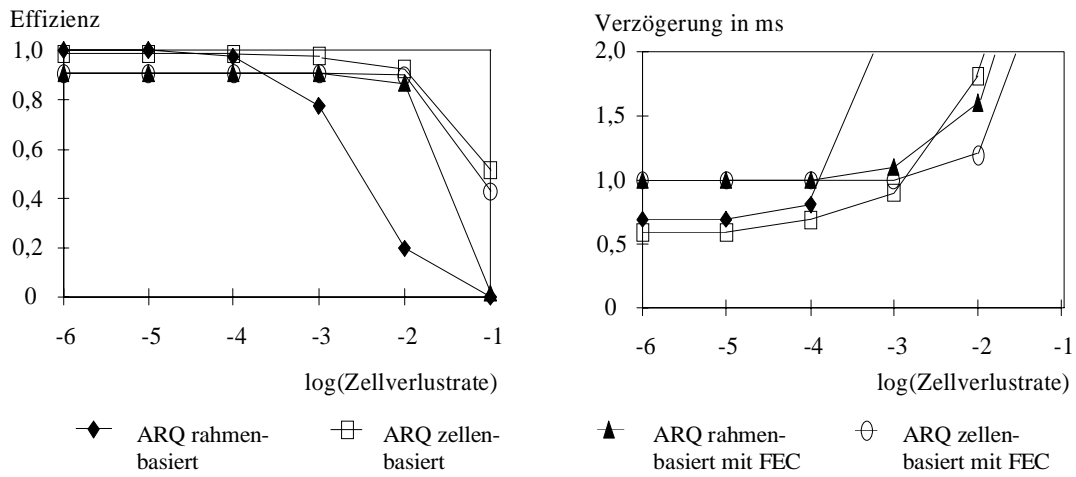


Abbildung 3. Simulationsergebnisse für Effizienz und mittlere Verzögerung

tionsserver voll belegen [CZ95]. Beim Einsatz von RMC-AAL kann gegenüber einem existierenden Vorschlag [Gol90], der zudem lediglich zuverlässige Punk-zu-Punkt-Kommunikation unterstützt, der zusätzliche Bandbreitenbedarf von 8,3% auf 2,1% gesenkt werden. Außerdem wird simulativ gezeigt, daß beim Einsatz von RMC-AAL gegenüber einem rahmenbasierten Verfahren in einem Szenario mit 16 Empfängern eine Ausnutzung der Netzressourcen von mindestens 95% auch bei einer mehr als 30fach größeren Zellverlustwahrscheinlichkeit ($10^{2,7}$ anstatt von $10^{4,2}$) möglich ist. Zur Simulation wurden mit dem Simulationswerkzeug BONEs (Block Oriented Network Simulator) detaillierte Modelle des Servers und der Endsysteme erstellt, die neben der Leistungsuntersuchung auch eine Validierung des entworfenen Systems ermöglichten [Dre94]. Eine umfängliche Leistungsbewertung der Kombinationen aus Vorwärtsfehlerkorrektur und Übertragungswiederholung erfolgte analytisch und simulativ, woraus Leistungsfunktionen für Durchsatz und Verzögerung ermittelt werden. Daraus werden Entscheidungskriterien für die Konfiguration der XAAL, für die optimale Dimensionierung der Rahmengröße und für den Umfang der Redundanz in Abhängigkeit von Empfängerzahl und Zellverlustwahrscheinlichkeit abgeleitet. Um auch den Realisierungsaufwand berücksichtigen zu können, wurden Kostenfunktionen für eine Implementierung von RMC-SSCS, RMC-AAL und des Gruppenkommunikationsservers mit Universalprozessoren und FEC-Hardwarebausteinen aufgestellt.

Abb. 3 zeigt Simulationsergebnisse für eine Gruppe mit 4 Empfängern und einer Entfernung von 100 km zwischen Sender und Empfängern sowie einer mittleren Datenrate von 100 Mbit/s. Die Simulationen ermöglichen einen Vergleich von RMC-SSCS und den drei Betriebsmodi von RMC-AAL bezüglich Effizienz (Verhältnis aus nutzbaren Zellen zur Gesamtzahl übertragener Zellen) und den mittleren Verzögerungszeiten. Für die Simulationen wurden eine Rahmengröße von 50 Zellen sowie zusätzliche 5 Redundanzzellen beim Einsatz von FEC verwendet. Die Simulationsergebnisse zeigen, daß für höhere Zellverlustwahrscheinlichkeiten eine zellenbasierte Übertragungswiederholung spürbare Vorteile gegenüber einer rahmenbasierten Übertragungswiederholung besitzt [Car95]. Die Simulationen zeigen ebenfalls, daß der Einsatz von FEC für einen weiten Bereich der Zellverlustwahrscheinlichkeit zu einer deutlichen Verringerung der Effizienz führt, jedoch Schwankungen der Auslieferungsverzögerung bis hin zu großen Zellverlustwahrscheinlichkeiten deutlich begrenzen können. Die Randbedingungen der Simulation führten beim Einsatz von FEC zu einer konstanten Grundverzögerung von 0.4 ms, was bei der vorliegenden Entfernung in ähnlicher Größenordnung wie die Übertragungsverzögerung liegt. Demgegenüber führen reine ARQ-Verfahren bei niedrigen Zellverlustraten und geringen Entfernungen zu einer geringeren mittleren Verzögerung. Mit wachsender Entfernung und wachsenden Zellverlustraten führt der Einsatz von FEC allerdings zu einer verringerten mittleren Verzögerung.

5 Zusammenfassung

Mit dem in dieser Arbeit entwickelten Gruppenkommunikationsserver und den neuen Protokollmechanismen für die Adaptionsschicht können zuverlässige Gruppenkommunikationsdienste in ATM-Netzen mit einer bisher nicht möglichen Dienstqualität und Effizienz erbracht werden. Die Ergebnisse für Leistungsfähigkeit und Implementierungsaufwand stellen umfassende Entscheidungskriterien für die Konfiguration und Dimensionierung der neuen ATM-Dienstkomponenten und für den Entwurf zukünftiger ATM-Gruppenkommunikationssysteme zur Verfügung.

ATM und IPng

Stefan Dresler

1 Einführung

In diesem Bericht möchte ich darlegen, in welche Richtungen sich meine persönliche Forschung bewegen könnte. Im wesentlichen sind dies der Bereich der ATM-Welt, das sich in der Entwicklung befindliche Internet-Protokoll der nächsten Generation, sowie mögliche Berührungspunkte beider. Konkret wird in dem Bericht zuerst die Verwendung von ATM in lokalen und öffentlichen Netzen angeschnitten. Anschließend wird auf die Historie des Entwurfs des neuen Internet-Protokolls eingegangen. Die Ausarbeitung schließt mit einigen Anmerkungen zum Betrieb von IPng über ATM.

2 Die Verwendung von ATM als Übertragungstechnologie

Mit ATM liegt eine Technologie vor, die zum Einsatz sowohl in LANs als auch für Weitverkehrsverbindungen geeignet ist.

2.1 ATM in lokalen Netzen

Wird ATM dazu benutzt, die Basis für ein lokales Netz bereitzustellen, so besteht mit [Lau94] ein Vorschlag, wie das weitverbreitete TCP/IP weiterhin verwendet werden kann. Dort wird vorgeschlagen, ein lokales ATM-Netz als *Logical IP Subnet (LIS)* zu konfigurieren. Kommunikation zu Rechnern im gleichen LIS erfolgt über ATM-Verbindungen, während für einen Datenaustausch mit Rechnern außerhalb dieses LIS ein IP-Router am „Rand“ des LIS verwendet werden muß. Ferner existiert ein mit diesem RFC kompatibler Mechanismus zur Kapselung von PDUs zur Versendung über AAL 5-Verbindungen [Hei93]. Die Unterstützung der erforderlichen Signalisierung ist Inhalt von [PLM⁺94]. Die genannten Dokumente sind im wesentlichen Ergebnisse der Arbeiten der *IP over ATM Working Group* der *IETF* (*Internet Engineering Task Force*).

2.2 ATM in öffentlichen Netzen

Im Weitverkehrsbereich ist ATM in Deutschland erst in wenigen Pilotprojekten realisiert bzw. geplant. Mit dem zunehmenden Angebot leistungsfähiger öffentlicher Netze wie ATM gewinnt die Idee der LAN-Kopplung über diese Netze an Attraktivität. Dabei werden räumlich entfernte LANs über ein öffentliches Netz gekoppelt (Abbildung 4).

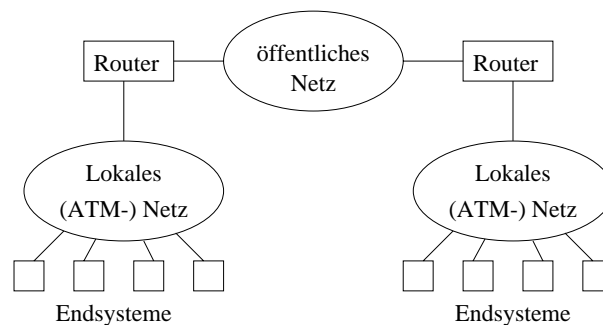


Abbildung 4. Kopplung von LANs über ein öffentliches Netz [Jef94]

Ein Konzept, das LAN-Kopplung realisiert, ist die sogenannte LAN-Emulation [Jef94] (Abbildung 5). Ein wesentliches Merkmal der LAN-Emulation ist es, daß auf beiden Seiten der Kommunikationsbeziehung diese Art der Kapselung unterstützt werden muß. Sie bietet den Vorteil, daß bestehende Implementierungen der Schichten bis herunter zur OSI-Schicht 2b (entsprechend IEEE 802.2) unverändert beibehalten werden können. Gegenüber einer direkten Verwendung von auf die ATM-Technik zugeschnittenen Schichtrealisierungen ist jedoch ein geringerer Durchsatz zu befürchten.

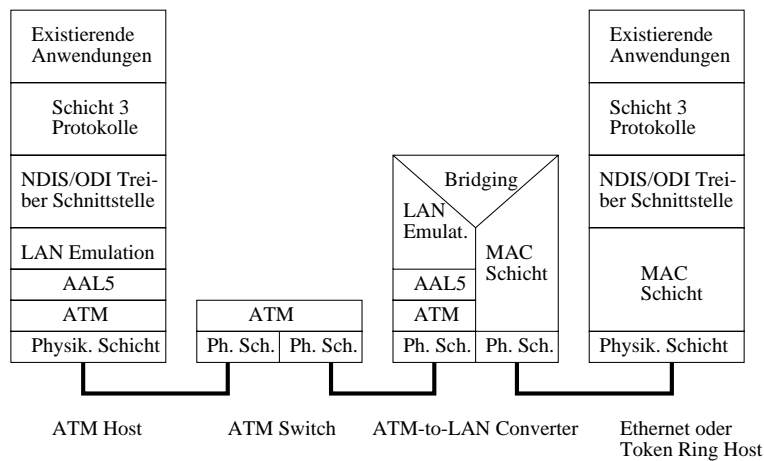


Abbildung 5. Beispiel für eine ATM LAN Emulation. ODI = Open Data-link Interface, NDIS = Network Device Interface Specification.

Wird nun sowohl im öffentlichen als auch im lokalen Netz ATM verwendet, so kann eine direkte Kopplung beider Netze Vorteile bringen. Dabei werden ATM-Verbindungen zwischen öffentlichem und lokalem Netz geschaltet. Besonders bei verbindungsorientierten Protokollen der Schichten 3 und 4, bei denen größere Datenmengen über den gleichen Pfad übertragen werden, kann dieses Vorgehen vorteilhaft sein. Inwieweit sich in der Praxis tatsächlich ein Vorteil zeigt, ist zu untersuchen.

3 Auswahl des Internet-Protokolls der nächsten Generation

Der Entwurf der TCP/IP-Protokollfamilie reicht etwa in die Jahre 1973/74 zurück [LR93, S. 9]. Zu dieser Zeit war der spätere weltweite Erfolg des auf TCP/IP aufbauenden Internets nicht vorhersehbar. Der zur Adressierung von Subnetzen und Endsystemen vorgesehene Adreßraum war für die damalige Zeit ausreichend, stellt sich heute jedoch als zu eng heraus. Dieser Adreßraum ist in verschiedene *Klassen* unterteilt. Organisationen bekommen auf Anfrage eine Adresse (oder mehrere) einer bestimmten Klasse zugeteilt. Durch dieses Verfahren entsteht ein gewisser Verschnitt, da in der Regel der Adreßraum einer Organisation nicht vollständig ausgenutzt wird. Bedingt durch die Klasseneinteilung und die nur teilweise Ausnutzung der zugeteilten Adreßräume werden die noch verfügbaren Adressen knapp, obwohl das gesamte Schema die Adressierung von bis zu $2^{32} \approx 4 \text{ Mrd.}$ Endsystemen erlaubt.

Ein weiteres Problem des derzeitigen Adressierungsschemas ist die stetig steigende Größe der Routingtabellen in IP-Routern. Das Verfahren des *Classless Inter-Domain Routing (CIDR)* [FLYV92] hat für eine kurzfristige Erleichterung gesorgt.

Bedingt durch die Verknappung der Adressen wird intensiv über einen Übergang vom derzeitigen Internet Protocol (genannt Version 4, IPv4) zum Internet Protocol der Version 6 (genannt *Internet Protocol – Next Generation, IPng*) nachgedacht. Da eine neuere diesbezügliche Studie eine Erschöpfung des Adreßraums „erst“ etwa in den Jahren 2005 bis 2011 voraussagt, bietet es sich an, eine Migration auf ein neues Protokoll dahingehend auszunutzen, auch andere Änderungen anzugehen.

3.1 Vorschläge zu IPng

Ende der 90'er Jahre wurde damit begonnen, verschiedene Vorschläge dazu auszuarbeiten, wie das Internet Protocol der nächsten Generation aussehen könnte. In Abbildung 6 findet sich die Historie der einzelnen Entwürfe. Die folgenden vier Vorschläge lagen dem IETF zur Beurteilung vor: *CATNIP*, *Nimrod*, *SIPP* und *TUBA*. *Nimrod* wurde eher als Studie denn als praktischer Vorschlag eingestuft, sodaß drei Entwürfe in die nähere Betrachtung kamen [Bra95a].

Allen Entwürfen ist gemein, daß sie als einen wesentlichen Punkt die Möglichkeit eines weiteren Wachstums des Internet durch die Verwendung eines größeren Adreßraums als IPv4 vorweisen.

Common Architecture for Next Generation IP (CATNIP) CATNIP [MU94] strebt die Integration von *CLNP* (OSIs *Connection-Less Network Protocol*), IP sowie Novells IPX an. Es soll die Verwendung von z.B. TP4, CLTP, TCP, UDP, IPX und SPX als Transportprotokolle über CLNP, IPv4,

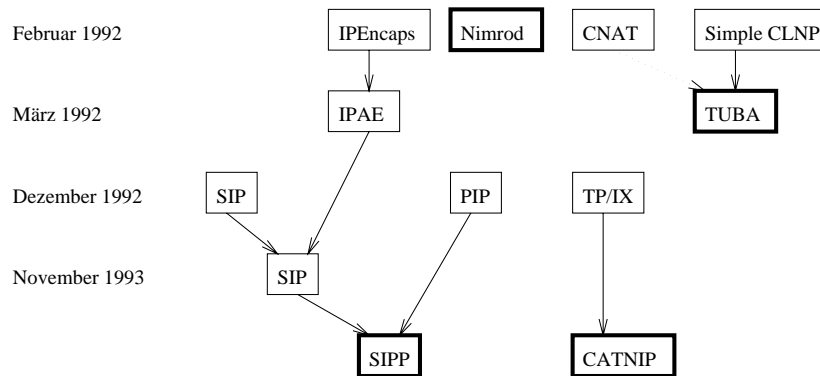


Abbildung 6. Historie der Entwürfe für IPng:

CATNIP = Common Architecture for a New Internet Protocol [MU94]

CLNP = Connection-Less Network Protocol (OSI)

IPAE = IP Address Encapsulation

PIP = The P Internet Protocol

SIP = Simple Internet Protocol

SIPP = Simple Internet Protocol Plus [Hin94b]

TP/IX = The Next Internet (auch Version 7 genannt) [Ull93]

TUBA = TCP and UDP with Bigger Addresses [Cal92]

IPX und CATNIP ermöglichen. Die Integration der verschiedenen Netzwerkprotokolle geschieht über eine komprimierte Form dieser Protokolle derart, daß gleiche Protocol Data Units (PDUs) verwendet werden. CATNIP verwendet Adressen gemäß OSI's Network Service Access Points (NSAPs) Format. Effizientes Routing soll durch geeignete Caching-Strategien gewährleistet werden. Diese tragen außerdem zur Reduzierung der Größe der Paketköpfe bei.

Viele von den beiden anderen Vorschlägen adressierten Aspekte sind bei CATNIP nicht zu finden. Es wurde daher bei der Evaluierung der verschiedenen Vorschläge allgemein als unfertig angesehen und schied deshalb als IPng aus. Dennoch weist es nach Ansicht der Reviewer viele innovative Konzepte auf.

Simple Internet Protocol Plus (SIPP) SIPP [Hin94b] ist als evolutionärer Schritt von IPv4 gedacht, der weitgehende Interoperabilität mit IPv4 gewährleistet. Dadurch wird der Übergangsprozeß von IPv4 zu IPv6 wesentlich vereinfacht. Merkmale des alten Protokoll, die sich bewährt haben, wurden übernommen; andere, die Probleme bereiten haben, wurden gestrichen oder ersetzt. Ferner ist SIPP auf einfache Erweiterbarkeit ausgelegt, sodaß spätere Ergänzungen etwa im Paketkopf durch das Konzept der *Extension Headers* möglich sind.

Der Entwurf von SIPP erwähnt explizit als zukünftige Schrittmacher der Entwicklung des Internet *Nomadic Personal Computing* (mobile Rechner), *Networked Entertainment* (Unterhaltungsindustrie, Video on Demand, etc.) und *Device Control* (Kontrolle von elektrischen Geräten des täglichen Lebens).

Es kommt dem Routing entgegen, indem es Cluster Adressen sowie Multicast Adressen zur Verfügung stellt. Zukünftige Multimedia-Anwendungen erhalten Unterstützung durch die Einführung eines sogenannten *Flow Labels*, mit dessen Hilfe eine gewisse Dienstqualität garantiert werden kann. Dazu kommt die Möglichkeit, Ende-zu-Ende-Verschlüsselung und -Authentisierung zu verwenden. Ein Autokonfigurationsmechanismus sorgt für Flexibilität bei der Vergabe von Adressen.

Der zur Beurteilung vorgelegte Entwurf zu SIPP sah eine Adressenlänge von 64 bit mit der Möglichkeit der Erweiterung um jeweils weitere 64 bit vor.

TCP and UDP with Bigger Addresses (TUBA) Der Entwurf für TUBA [Cal92] sieht vor, nur IP durch das *Connection-Less Network Protocol (CLNP, ISO/IEC 8473)* zu ersetzen. Entsprechend steht der größere Adreßraum der Network Service Access Points (NSAPs) zur Verfügung. TUBA erfordert weder Kapselung noch Übersetzung von Paketen noch eine Adreßabbildung. In der Migrationphase kann es unabhängig von IP verwendet werden. Routing und Weiterleiten von IP und CLNP-Paketen geschieht unabhängig voneinander.

3.2 Beurteilung

Die drei Vorschläge wurden beurteilt und miteinander verglichen. Dabei wiesen alle Entwürfe Schwächen auf. Es zeigte sich jedoch, daß mit gewissen Änderungen SIPP und TUBA als geeignet erschienen.

Nach der Diskussion wurden die kritischen Punkte in SIPP überarbeitet. Es weist jetzt ein fixes Adressformat von 128 bit auf, das als Kompromiß zwischen Effizienz, Funktionalität, Flexibilität und globaler Anwendbarkeit angesehen wird. Dieser überarbeitete Entwurf wird nun von der Network Working Group der IETF als Basis für IPng empfohlen [Hin94a, Hin95, Bra95a]. Er löst die meisten der erkannten Probleme, insbesondere auf den Gebieten der Adressierung, des Routings, des Migrationsprozesses und der automatischen Adreßkonfiguration.

4 IPng über ATM

Gegenüber dem derzeitigen Internet Protocol der Version 4 erlaubt IPng die Verwendung eines *Flow Labels*. Dieses signalisiert den an der Übertragung eines Pakets beteiligten Knoten, daß das Paket eine spezielle Behandlung erfahren soll. Dies kann z.B. eine Verarbeitung innerhalb einer anzugebenden maximalen Zeit sein, sodaß auf diese Weise einer Anwendung eine gewisse Dienstgarantie gegeben werden kann. Hier bietet sich die Verwendung von IPng in Verbindung mit einem Ressourcen-Reservierungsprotokoll wie RSVP [ZDE⁺] an. Die Kombination von IPng und RSVP kann dann über ATM gesetzt werden. Die genaue Protokollarchitektur sowie Fragen der Signalisierung bilden einen interessanten Aspekt, den es noch zu klären gilt.

Abbildung 7. ITEMS: Softwaretechnik für kooperative, multimodale und ubiquitär verfügbare Anwendungen

1 Kooperation

1.1 Taxonomie und Koordinierungstheorie

Die Entwicklung einer Kooperationstaxonomie wurde von einer Reihe verschiedener Forschungsgebiete angegangen. Frühe Definitionen aus dem Bereich *Computer Supported Cooperative Work (CSCW)* oder auch *Workgroup Computing* spannen eine Taxonomie entlang der Achsen Zeit und Raum auf. Es hat sich jedoch gezeigt, daß zum einen gerade im unklar definierten Bereich von ‘beliebige Zeit, beliebiger Ort’ die aus Anwendersicht relevanten Systeme liegen und zum anderen durch die Entwicklungen im Bereich verteilte Systeme mit der annähernd erreichten Verteilungstransparenz die Ort-Achse an Bedeutung verloren hat.

Interessanter sind die Ansätze, die im betriebswirtschaftlichen Umfeld entwickelt wurden. Eine Kooperation besteht hier aus den Elementen [Jab95]

Ziel als das Ergebnis allen Handelns

Aktoren als agierende Subjekte (durchaus auch automatisiert)

Aktion als die elementaren Handlungseinheiten

Abhängigkeiten, die bei der Ausführung der Aktionen durch die Aktoren zu beachten sind.

Der Entwurf eines Anwendungssystems, das eine Gruppe kooperierender Anwender unterstützt, umfaßt analog dazu eine Reihe von Teilschritten. Zuerst muß das *Ziel* der Kooperation bestimmt werden, das dann in Teilziele unterteilt werden kann, aus denen sich schließlich die *Aktionen* der Anwendung ableiten lassen. Aktionen sind dabei in der Regel Rechenprozesse oder Benutzeraktionen, die - ähnlich einem Prozeduraufruf - unter vorgegebenen Bedingungen ein definiertes Ziel erreichen. Diesen werden die Aktoren, also die Anwender des Anwendungssystems, zugeordnet. In der Regel werden beim Ausführen der Aktionen die Aktoren in irgend einer Art interagieren (sei es durch direkten Informationsaustausch oder durch verwenden der gleichen Ressourcen), so daß *Abhängigkeiten* zwischen den Aktoren bestehen. Diese Abhängigkeiten führen, insbesondere da alle Aktoren dem selben Ziel zustreben, fast immer zu einer Art Konflikt oder Konkurrenz, die das Anwendungssystem *koordiniert* muß. Während nun Ziel, Teilaktivitäten und Aktoren beim Entwurf einer Anwendung relativ leicht erfasst, modelliert und zur Laufzeit gemanaged werden kann, erweist sich eine Automatisierung und Rechnerunterstützung der Koordination als ungleich komplexer.

1.2 Kooperationstypen und ihre Koordination

Eine Koordination kooperierender Aktoren ist stark abhängig von der Art der Kooperation. Relevante Kooperationsaspekte und ihre Auswirkungen auf die Koordination sind [Syr93]

- **zeitliche Verteilung** der Aktionen im Spektrum synchron (enger zeitlicher Bezug z.B. in einer Konferenzsituation) und asynchron (keine engen Zeitbeziehung zwischen den einzelnen Aktionen, typisch für Workflow-Systeme und WWW). Durch die enge Kopplung der Aktoren bei einer synchronen Kooperation besteht viel Platz für eine implizite Koordination durch ein soziales Protokoll. Asynchrone Kooperation ist dagegen durch die starke Entkoppelung auf eine explizite Koordination durch das Anwendungssystem angewiesen
- **Gruppengröße** zwischen kleinen Gruppen, die sich z.T. selber koordinieren können, und großen Gruppen, die vom Anwendungssystem im Einklang mit der ebenfalls aktiven Rahmenorganisation koordiniert werden. Eng damit verbunden ist der **organisatorische Bezug** einer Aktion, die sich nur auf die Gruppe erstrecken kann oder eine ganze Organisation umfassen kann
- **Strukturierungsgrad** einer Kooperation im Zusammenhang mit der **Wiederholfrequenz**. Großes Wissen über die Struktur einer Aktivität ist Voraussetzung für eine starke explizite Koordination durch das Anwendungssystem, die wiederum Notwendig für eine Automatisierung ist
- **Koordinationsziel** zwischen Effizienz und Flexibilität, denn strikte Koordination kann zwar die Effizienz durch klare Richtungsvorgaben erhöhen, schränkt aber die Flexibilität ein.

Bei einer konkreten Kooperation können diese einzelnen Aspekte in beliebiger Zusammenstellung verschiedene Koordinationsformen verlangen. Nur wenige Kooperationsformen lassen sich hier in ein Schema pressen. Eine Automatisierung der Koordination in kooperativen Anwendungssystemen wird jedoch von den Anwendern nur dann akzeptiert, wenn sich die Koordination an die jeweils notwendige Kooperationsform anpaßt.

2 Kooperationsmodellierung

Das Kernproblem beim Entwurf kooperativer Anwendungssystemen liegt also in der geeigneten Modellierung einer Kooperationsform, für die dann eine adäquate Koordination ausgewählt wird. Der Ansatz, der dafür in ITEMS gewählt wurde, beruht auf den Metaphern *Arbeitsszenario* und *Szenario-Fluß*.

2.1 Modellierung primär synchroner Kooperation

Ein Arbeitsszenario dient zur Modellierung einer relativ stabilen Situation einer Kooperation. Zum Aufbau eines Arbeitsszenarios stehen die Basiselemente 'Benutzer' (als Repräsentation einer Benutzungsschnittstelle), 'Agent' (als Repräsentation von Prozessen) und 'Archiv' (als Abstraktion für persistente Daten) und Verfeinerungen davon zur Verfügung (siehe Abb. 8). Diese Elemente dienen analog zum objektorientierten Ansatz zur Beschreibung von Aktoren und Aktionen.

Abbildung 9. Aspekte eines Arbeitsszenarios in Fragmenten beschrieben

2.3 Asynchrone Formen: Situationswechsel

Situationswechsel entstehen durch Ereignisse, die zu weitgehend anders strukturierten Arbeitsszenarien führen. Diese Ereignisse treten oft spontan auf und ändern eine Situation für eine gewisse Zeit so

Abbildung 11. Modellierung stark entkoppelter Einzelschritte in Abläufen

3 Bezug zu realen Systeme

Mit dem vorgestellten Modellierungsmetaphern wurde das Ziel verfolgt, auf der Basis elementarer Koordinationsbausteine komplexe kooperative Anwendungen zu entwerfen. Entscheidend für die Anwendbarkeit und Realitätsnähe (und damit auch die Benutzerakzeptanz) ist dabei die richtige Auswahl dieser Koordinationsbausteine. Dazu werden zur Zeit eine Reihe Anwendungen mit unterschiedlichen Kooperationsformen untersucht, angefangen von der stark synchronen, konferenzbasierten Kooperationsform des BERKOM Teledienstes MMC [Fri94] über die Verwendung dieses Teledienstes als Baustein in einem Tele-Teaching Szenario bis hin zu Kooperationsmischformen wie im Kundeninformationssystem für SAP, daß asynchrone Interaktion über WWW mit synchronen Konferenzszenarien koppelt. Ziel der Analyse dieser Systeme ist die Identifikation der zugrundeliegenden Kooperationsformen, die Isolation der darin verwendeten Koordinationsmechanismen und schließlich ihre Abbildung in Koordinationsbausteine. Besondere Beachtung finden dabei die Übergänge von einer Kooperationsform in eine andere innerhalb eines Anwendungssystems und ihre Abbildbarkeit in eine der vorgestellten asynchronen Modellierungsformen.

Softwaretechnik für flexible und multimodale interaktive Anwendungen

Hans-Werner Gellersen

1 Einleitung

Durch die rasante Entwicklung von Interaktionstechnologien entstehen ständig neue Möglichkeiten, die Mensch-Computer-Interaktion (MCI) in interaktiver Software zu realisieren. Zugleich entstehen durch Erschließung neuer Anwendungsbereiche und Anwendergruppen sehr verschiedenartige Anforderungen an die Realisierung der MCI. Ausgehend vom Technologieschub und vom Anwendungsbedarf ist also eine Diversifikation der MCI in interaktiven Anwendungen zu erwarten. Mit anderen Worten: es ist zu erwarten, daß für ein und dieselbe Anwendung verschiedenartige MCI-Komponenten zu entwickeln sind. Der Begriff MCI-Komponenten umfaßt dabei alle Komponenten, die MCI-Entwurfsentscheidungen einkapseln und wird hier zur Vermeidung des ungenauen und meist mißverständlich benutzten Begriffs *Benutzerschnittstelle* verwendet.

Angesichts der Diversifikation der MCI in interaktiven Anwendungen ist die Softwaretechnik gefordert, Flexibilität zu unterstützen. Flexibilität interaktiver Software ist die Grundvoraussetzung für eine Anpassung an verschiedene Interaktionstechnologien und verschiedene Anwendungsanforderungen. Die Flexibilität soll dabei nicht nur Anpassung im Sinne von Individualisierung (*Customization*) erlauben, sondern vielmehr Anpassung an verfügbare Medien, neu entstehende Interaktionstechniken und sogar verschiedene Interaktionsmetaphern¹. Die Softwaretechnik ist also gefordert, durch eine systematische Vorgehensweise interaktive Software derart zu partitionieren, daß das geforderte Maß an Flexibilität bei gleichzeitiger Maximierung der Wiederverwendung von Komponenten erreicht wird. Grundlagen dazu werden in Abschnitt 2 diskutiert.

Der technologische Fortschritt im Bereich der MCI führt nicht nur zu einer Diversifikation der MCI in interaktiven Anwendungen, sondern wirft zusätzlich das Problem der Auswahl und Kombination geeigneter Interaktionsmedien, -techniken und -metaphern auf. Eine konkrete Art und Weise der MCI, die sich aus der Wahl einer bestimmten Interaktionstechnik ergibt, wird als Modalität bezeichnet. Die Problematik der Auswahl und geeigneten Kombination von Modalitäten wird entsprechend unter dem Begriff Multimodalität behandelt. In Abschnitt 3 wird multimodale MCI mit Bezug auf verschiedene Ebenen im Entwurf interaktiver Software diskutiert.

2 Softwaretechnik für interaktive Anwendungen

2.1 Strukturierung von Entwurfsentscheidungen

Der Begriff *Benutzerschnittstelle* als Eindeutschung des Begriffs *User Interface* importiert die begriffliche Ungenauigkeit des Originals. In der deutschen Fachliteratur werden daher die Begriffe *Benutzungsschnittstelle* und *Bedienoberfläche* unterschieden. Ersterer beschreibt das *was* der MCI, letzterer das *wie*. Diese Unterscheidung ist sehr wesentlich, aber doch zu grob zur Klassifizierung von Entwurfsentscheidungen in interaktiven Anwendungen.

In diesem Abschnitt wird ein Rahmen zur Klassifikation von MCI-Entwurfsentscheidungen vorgestellt, der hier als Grundlage für die Diskussion softwaretechnischer Aspekte dienen soll, darüberhinaus aber auch ein erster Schritt hin zu einer Methode für interaktive Anwendungen ist. Die wichtigsten Anforderungen an einen solchen Rahmen sind Vollständigkeit, Trennbarkeit von Entscheidungen, Minimierung von Abhängigkeiten über Klassengrenzen sowie Wiederverwendbarkeit.

Ein klassischer Entwurfsrahmen für interaktive Software ist das linguistische Modell von Foley und van Dam [FvD90], das eine Aufteilung in konzeptionelle, semantische, syntaktische und lexikalische Entscheidungen vorschlägt. Diese Aufteilung in Verarbeitungsstufen natürlicher Sprache basiert auf der Sichtweise, daß MCI-Komponenten Übersetzer zwischen Mensch und Computer sind. In der Praxis erweisen sich diese Analogie und so auch der Entwurfsrahmen jedoch als weniger geeignet, da sie eine Pipelinestruktur für MCI-Komponenten suggerieren. In [Gel95] und [Gel94a] wird ein Entwurfsrahmen vorgestellt, der insbesondere durch MCI-Entwurfsentscheidungen in telekooperativen Anwendungen

¹ Interaktionsmetaphern sind Metaphern für die Art und Weise der MCI und müssen von Metaphern für Aufgabengebiete unterschieden werden [Hut89]. Konversation ist z.B. eine Interaktionsmetapher, während Desktop oder Office Aufgabengebietsmetaphern sind.

motiviert ist. Als Weiterentwicklung und Vereinfachung dieses Strukturierungsansatzes werden hier die folgenden Klassen von MCI-Entwurfsentscheidungen vorgeschlagen:

1. Konzeptionelles Modell
2. Dialog
3. Oberfläche
4. Pragmatik

Das konzeptionelle Modell einer interaktiven Anwendung legt das Verständnis des Anwenders für eine Anwendung fest, indem es die konzeptionellen Objekte, Beziehungen und Abhängigkeiten zwischen Objekten sowie Anwenderfunktionen auf den Objekten spezifiziert. Entscheidungen in dieser Klasse sind im wahrsten Sinne des Wortes entscheidend für die Benutzbarkeit einer Anwendung. Eine weitere Auftrennung dieser Klasse nach strukturellen und funktionalen Entwurfsaspekten ist ebenfalls denkbar, da diesbzgl. eine klare Trennung gegeben ist.

Die Dialogklasse umfaßt Entscheidungen zum Inhalt und zur Reihenfolge des Informationsaustausches zwischen Anwendung und Anwender. Kernfragen in dieser Klasse sind: Welche Informationseinheiten (semantische Token) werden zwischen Anwender und Anwendung ausgetauscht? Wie werden diese Einheiten zu Nachrichten strukturiert? Welchen Abfolgen des Nachrichtenaustausches sind sinnvoll?

Die Oberflächenklasse umfaßt alle Entscheidungen, die die unmittelbare Interaktion zwischen Anwender und Anwendung an der Bedienoberfläche betreffen. In dieser Klasse werden Interaktionsobjekte spezifiziert, die aus Entwicklersicht Softwaremodule sind, die die Ein-/Ausgabe mit dem Benutzer durchführen. Aus Anwendersicht sind es Objekte, die Informationen vom System vermitteln, und die manipuliert werden können, um dem System Information zu übermitteln (Austausch lexikalischer Token). Entscheidungen zur Pragmatik der MCI betreffen die Hardware- und Systemumgebung einer interaktiven Anwendung, sowie weitere Randbedingungen. Kernaspekt ist die Ergonomie von Ein-/Ausgabegeräten. Unterstützende Software wie Gerätetreiber und Fenstersysteme kann ebenfalls dieser Klasse zugerechnet werden, sofern sie als gegebener Teil einer Umgebung und nicht als Option der Oberflächengestaltung aufgefaßt wird.

2.2 Modelle, Methoden, Architekturen

Für die Entwicklung interaktiver Software sind zahlreiche Modelle, Methoden und Architekturen vorgeschlagen worden. Die uneinheitliche Terminologie macht es hier sehr schwierig, existierende Ansätze zu vergleichen. Im folgenden wird ausgehend vom Begriff der Softwarearchitektur der Bezug zwischen Architektur und Modell bzw. Architektur und Methode hergeleitet.

Eine Softwarearchitektur beschreibt die Aufteilung von Software in Komponenten festgelegter Funktionalität sowie die Beziehungen zwischen den Komponenten. Modelle stellen Abstraktionen von Architekturen dar und werden oft auch als konzeptionelle Architekturen bezeichnet. Modelle definieren ebenfalls Komponenten mit bestimmter Rolle in einer Software, abstrahieren aber i. allg. von Topologien und von Komponentenschnittstellen [Coc90]. Methoden schließlich teilen einen Entwurf in einzelne Schritte auf und identifizieren die Entscheidungen, die auf jeder Ebene zu fällen sind. Architekturen können als Entwurfsmethoden aufgefaßt werden, da sie durch Vorgabe von Komponenten eine strukturierte Vorgehensweise implizieren. Insbesondere implizieren Architekturen die Orientierung einer Methode, d.h. sie legen fest, ob ein Entwurf Top-Down, Bottom-Up oder Middle-Out erfolgt. Während bei einer Methode aber von Entwurf zu Entwurf nur die Methode selbst wiederverwendbar ist, bietet eine Architektur eine wiederverwendbare Softwaredekomposition [Car92].

Zu den bekanntesten Modellen interaktiver Software gehören das Seeheim-Modell [Pfa85], MVC [KP88] und PAC [Cou87]. Das Seeheim-Modell ist ein Schichtenmodell, das interaktive Software in die Komponenten Anwendungsschnittstelle, Dialogkontrolle und Präsentation aufteilt, die den Klassen 1 bis 3 im oben vorgestellten Entwurfsrahmen weitgehend entsprechen. PAC ist ein homogenes objektorientiertes Modell, wobei jedes Objekt intern Seeheim-konform organisiert ist. MVC ist ebenfalls objektorientiert, dreiteilt aber die Objektwelt in Model-, View- und Controllerobjekte. Diese Dreiteilung weicht etwas vom Seeheim-Modell ab, indem die Entscheidungen der Klasse 3, die Anwendereingaben betreffen, zusammen mit Dialogentscheidungen in Controllerobjekten eingekapselt werden, während Viewobjekte nur Entscheidungen bzgl. Systemausgaben modellieren.

Neben solchen Architekturen, die direkt aus diesen Modellen interaktiver Software abgeleitet sind, findet man heute vorwiegend sehr viel einfachere Architekturen interaktiver Anwendungen. In monolithischen Anwendungen sind Entwurfsentscheidungen der Klassen 1 bis 3 in einer Komponente zusammengefaßt, nur Entscheidungen zur Pragmatik sind separat implementiert. In Client-Server-Anwendungen finden sich hingegen zwei Hauptkomponenten: der Client implementiert konzeptionelles Modell und

Abbildung 12. Architekturen interaktiver Anwendungen

Dialog, während der Server die Darstellung implementiert. Abbildung 12 gibt einen Überblick über Architekturen interaktiver Software.

2.3 Stand der Entwicklung

Der Stand der Entwicklung interaktiver Anwendungen ist geprägt durch sog. GUI-Anwendungen. Diese Anwendungen bauen auf einer Client-Server-Architektur auf, in der der Server die Oberfläche implementiert. Bausteine dieser Architektur sind einfache Komponenten für Bedienoberflächen (sog. GUI-Toolkits), wodurch eine Bottom-Up-Orientierung des Entwurfs impliziert wird. Wie bei allen Architekturen des Bottom-Up-Typs ist es möglich, sukzessive neue Komponenten zu konstruieren, die Architektur bietet dafür jedoch keine methodische Unterstützung. Die Architektur fördert eine Wiederverwendung von Komponenten der Bedienoberfläche für verschiedene Anwendungen. Umgekehrt wird eine Wiederverwendung der von der Oberfläche unabhängigen Teile einer Anwendung, die ja Grundvoraussetzung für die in der Einleitung motivierte Flexibilität ist, nicht unterstützt.

Neben GUI-Toolkits existieren eine Vielzahl von User Interface Management Systemen (UIMS). Ältere Ansätze basieren i. allg. auf einer starren Seeheim-Architektur mit reihenfolgeorientierten Dialogsteuerungen. Neuere Ansätze sind schlichtweg um Werkzeuge zur Teilautomatisierung des Entwurfs erweiterte GUI-Toolkits, in denen der Anspruch expliziter Dialogsteuerung mit entsprechender Konsequenz für die Wiederverwendbarkeit aufgegeben wird.

3 Multimodale Mensch-Computer-Interaktion

3.1 Terminologie

In der MCI-Literatur wird der Begriff der Modalität oft mit dem des Modus verwechselt. In einem System mit mehreren *Modi* kann ein und dieselbe Interaktion verschiedene Effekte haben. In einem System mit mehreren *Modalitäten* kann hingegen ein und derselbe Effekt durch verschiedene Arten der Interaktion erzielt werden.

Eine Interaktionsmodalität bezeichnet also eine Art und Weise der MCI. Sie ist festgelegt durch die folgenden Komponenten:

- Kommunikationskanal des Menschen (Sensor- und Aktorkanäle)
- Kommunikationskanal des Computers (Ein-/Ausgabegeräte)
- Darstellungssystem auf lexikalischer Ebene (Medien)
- Dialogstil

Die möglichen Konfigurationen dieser Komponenten fallen in zwei grundlegende Modalitätenkategorien: Sprache und Aktion. Sprachbasierte oder besser linguistische Modalitäten nutzen vor allem die Abstraktionsmöglichkeiten in Sprache, die sehr mächtige und fokussierte Ausdrücke erlauben. Sprachbasierte Modalitäten sind jedoch indirekt, sie erfordern einen Interpretationsschritt zur Ableitung von Handlungen aus Worten. Aktionsbasierte Modalitäten hingegen sind direkt, durch Aktion wird die Welt direkt

verändert. Aktionsbasierte Modalitäten sind in Bezug auf eine Interaktion sehr spezifisch (alle Teilaspekte werden dargestellt), in hohem Maße analog (Analogie zwischen Darstellung einer Information und der tatsächlichen Information), aber unfokussiert (keine Bezugnahme auf einzelne Aspekte) [Ber93].

In multimodaler MCI werden mehrere Modalitäten kombiniert, wobei die Kopplung verschieden eng sein kann (alternativ, synergetisch, parallel oder verschmolzen). Von Multimodalität spricht man insbesondere dann, wenn durch die Kombination verschiedene Kommunikationskanäle des Menschen genutzt werden. Eine solche Einschränkung mag für kognitionspsychologische Betrachtungen sinnvoll sein, für die softwaretechnische Diskussion wird jedoch die weitergehende Definition bevorzugt.

3.2 Softwaretechnische Aspekte

Multimodalität betrifft alle in Abschnitt 2.1 beschriebenen Klassen von Entwurfsentscheidungen. Pragmatische Entscheidungen legen die verfügbaren Kommunikationskanäle fest und bestimmen dadurch den Entwurfsraum für multimodale interaktiven Software.

Die Oberflächenklasse definiert Objekte für die Interaktion mit dem Anwender. Diese Objekte können als Signal-, Auswahl-, Editier- und zusammengesetzte Objekte klassifiziert werden. Alle Nachrichten des Systems an den Anwender werden durch Signalobjekte abgewickelt. Benutzeraktionen werden entweder direkt auf Anwendungsfunktionen abgebildet (Signalobjekte), oder dienen der Selektion aus einer Auswahl (Auswahlobjekte), oder manipulieren Editierobjekte auf lexikalischer Ebene. Alle diese Objekte sind modalitätsabhängig. Durch Zugehörigkeit zu einer Modalität erben sie Eigenschaften, die ihre Eignung für die Realisierung verschiedener Interaktionsziele bestimmt. Eine offene Frage ist, ob es bereits auf dieser Ebene multimodale Objekte gibt, oder ob Multimodalität erst durch Verknüpfung lexikalischer Token verschiedener Objekte entsteht. Letzteres ist attraktiver im Sinne einer softwaretechnischen Modularisierung, es setzt aber die getrennte Erzeugung lexikalischer Token voraus, was z.B. bei Verwendung von Neuronalen Netzen zur Verknüpfung multimodaler Signale nicht gegeben ist.

Existierende Ansätze multimodaler MCI konzentrieren sich auf die Oberflächenebene, sind aber getrennt nach Ein- und Ausgabe. Multimodale Eingabe wird primär als Erweiterung natürlich-sprachlicher Interaktion untersucht und ist durch zwischenmenschliche Kommunikationsformen motiviert. Multimodale Ausgabe oder auch *Intelligente Multimedia-Darstellung* untersucht die automatische Generierung von Darstellungen verschiedener Modalität ausgehend von abstrakten internen Informationsrepräsentationen.

Auf der Dialogebene wird der Informationsfluß zwischen Anwender und Anwendung beschrieben. Als zugrundeliegende Struktur werden dafür Dialogstile herangezogen, die wiederum auf Interaktionsmetaphern basieren. Eine solche Metapher ist z.B. die Konversation, die kommando- und natürlichsprachlichen Dialogen zugrundeliegt. Verschiedene Dialogstile weisen unterschiedliche Eigenschaften hinsichtlich der Vermittlung von Information auf, wobei die Unterschiede im wesentlichen wieder auf die Kategorien Sprache und Aktion zurückgehen. Durch geeignete Kombination von Dialogstilen, d.h. durch Multimodalität auf der Dialogebene, kann die Informationsvermittlung verbessert werden. Ein bekanntes Beispiel hierfür ist die Erweiterung direktmanipulativer Dialoge um sprachliche Komponenten in Form von Menüs oder Dialogboxen.

Auch hinsichtlich des konzeptionellen Modells ist die Wahl geeigneter Modalitäten relevant, da die Eigenschaften von Darstellungsmodalitäten die semantische Distanz zwischen Modell und Darstellung bestimmen. Das heißt, schon bei der Erstellung eines konzeptionellen Modells spielen technologische Möglichkeiten der MCI eine Rolle.

4 Ausblick: methodische Unterstützung für Flexibilität und Multimodalität

Im Rahmen dieses Berichts wurde Softwaretechnik für interaktive Systeme mit Blick auf Flexibilität und Multimodalität andiskutiert. An dieser Stelle verbleibt nur Raum für einige grundlegende Überlegungen zur methodischen Unterstützung.

Flexibilität ist ein Architekturproblem, das weniger an der Identifikation der Komponenten krankt (hier ist das Seeheim-Modell durchaus richtungsweisend), sondern vielmehr an der Definition der Schnittstellen. In heutigen Client/Server-Architekturen beispielsweise ist die Schnittstelle sehr serverspezifisch was Clients praktisch nicht wiederverwendbar macht.

Multimodalität setzt eine flexible Softwarearchitektur voraus, ist aber kein Architekturproblem als solches. Die Problemstellung liegt hier vielmehr in der methodischen Unterstützung des Ausfüllens von Komponenten sowohl auf Oberflächen- als auch auf Dialogebene. Ansatzpunkt hierfür ist die Entwicklung einer Modalitätentheorie dergestalt, daß Aussagen über die semantische Distanz einer Modalität zu einem konzeptionellen Modell möglich werden.

Abbildung 13. 3-D Modell elementarer Mobilitätsklassen

2.1 Informationen in mobiler Umgebung

Unter dem Begriff *Informationsobjekte* sind sowohl Daten wie auch (Informations-) Dienste zusammengefaßt. Diese können durch verschiedene Kriterien attribuiert werden:

1. Lokation

Unter diesem Aspekt ist nicht die Lokation der Information angesprochen, sondern vielmehr die Lokation des Benutzers, der auf eine Information zugreift. Allgemein üblich sind lokationsunabhängige Zugriffe auf bestimmte Datenobjekte. Als Beispiel dafür sei eine Kundenstammdatei im Versicherungswesen betrachtet. Unabhängig vom Ort eines Vertreters greift dieser überall auf ein und dieselbe Datei zu. Dabei spielt es für den Vertreter keine Rolle, ob es sich um den Originaldatensatz oder um ein von der Netzverwaltung erstelltes Replikat handelt.

In Zukunft werden jedoch zunehmend die Informationszugriffe eine Rolle spielen, die entscheidend vom Ort des Zugriffs abhängen. Als Beispiel dient hier ein Szenario aus dem Freizeitbereich: Ein Reisender will sich über das lokale Kinoprogramm informieren. Er sollte unabhängig von seiner Position dieses Datum unter einem einheitlichen Bezeichner anwählen können. Die Netzverwaltung bildet diesen Bezeichner nun abhängig von Zugriffsort auf den entsprechenden lokalen Service ab. Ansätze hierzu finden sich bei den Mehrwertdiensten, wie sie im Rahmen der *Intelligent Networks* diskutiert werden [DV92].

2. Benutzer

Der Zugriff auf Informationsobjekte, darunter in erste Linie Daten, können benutzerabhängig sein. Als Beispiel sei hier ein Mail-Datensatz genannt. Ein Mail-Datensatz kann erst durch die Angabe des Benutzernamens korrekt adressiert werden. Die Adreßabbildung geschieht unabhängig vom Ort des Zugriffs und vom Ort des Datensatzes..

3. Information

Informationsobjekte können in Relation zu anderen Informationsobjekten stehen. Ein Dokument z.B. impliziert das Vorhandensein von Schriftsatzobjekten, ein Dienst kann auf Ressource-Dateien zurückgreifen müssen.

4. Gerät

Repräsentationsobjekte von Informationen sind stark gerätebezogen. Objekte mit Farbinformation machen auf 1-Bit-Graphikschirmen wenig Sinn. Ebenso ist eine Transformation von MM-Daten für eine Darstellung über Nicht-MM-Geräten notwendig.

5. Verbindungskosten

Entscheidend für die Auswahl eines Informationsobjekt sind die bei einer Übertragung anfallenden Verbindungskosten. Darunter fallen u.a. *Zeit* (wichtig bei schmalbandigen Verbindungen) und *finanzielle Kosten* (wichtig bei WAN-Verbindungen). Bei schmalbandiger Verbindung oder bei hohen Leitungskosten kann z.B. eine automatische Farb- und Bildfrequenzreduktion bei einer Videoübertragung notwendig sein.

2.2 Benutzer in mobiler Umgebung

Auch für den Zugriff auf bzw. die Adressierung von Personen können ähnliche Aspekte identifiziert werden. Ergänzend ist hier vor allem das Attribut **Rolle** erwähnenswert. Die Adressierung einer Person mit der Rolle *Chef* wird für einen Benutzer immer und überall gleich aussehen. Hat die Person aber eine allgemeingültigere Rolle, wie z.B. *Notarzt*, so macht es Sinn, diese durch zusätzliche Attribute *der nächste*, *der billigste* etc. zu klassifizieren.

2.3 Geräte in mobiler Umgebung

Schließlich sind diese Überlegungen auf Geräte und Gerätfragmente abbildbar. Der Rechner einer bestimmten Person oder Firma ist lokationsunabhängig, während die Identifikation des *nächstgelegenen* PostScript-Drucker sehr wohl von der Lokation des Zugriffs abhängt. Als Gerät sind nicht etwa nur komplette Rechner zu modellieren, auch sog. *Gerätefragmente* wie z.B. PCMCIA-Karten werden eine zunehmend unabhängige Rolle spielen.

3 Mobiler Informationszugriff

In den vorigen Kapiteln wurden das Zusammenspiel von Benutzer, Information und Gerät beim mobilen Rechnereinsatz unter verschiedenen Gesichtspunkten vorgestellt. Beispielhaft wird im folgenden eine Anwendung aus dem Bereich des mobile Computing besprochen und eine Realisierung vorgestellt.

Abbildung 14. Terminal vs. Workstation Dilemma

Bei der vorliegenden Realisierung wurde ersteres Prinzip gewählt. Dadurch waren die softwaretechnischen Implikationen anschaulicher zu bearbeiten und die daraus resultierende Anwendung entsprach den Vorgaben, einen möglichst unabhängigen WWW-Client zu entwickeln. Der Aufbau ist in Abb. 15 illustriert.

Netzzugang und Dienstzugriff Als Netzzugang dienen für WWW TCP/IP-Verbindungen. Für den PDA ist aber keine entsprechende Protokollrealisierung verfügbar. Daher wurde eine Workstation als Gateway zum Internet (TCP/IP) zwischengeschaltet. Die Verbindung zur Workstation wird wahlweise über eine Modemverbindung oder über eine serielle Leitung mit proprietärem Übertragungsprotokoll aufgebaut.

Der Servicezugriff wird beim WWW durch das http-Protokoll vorgenommen. Bei der Entwicklung konnte die Protokollverwaltung auf dem PDA realisiert werden.

Abbildung 15. verteilte Klientenarchitektur des Pda-WWW-Browser

4 Zusammenfassung

Softwareentwicklung für Anwendungen in mobiler Umgebung erfordert eine erweitertere Betrachtungsweise als heute unter dem Stichwort *mobile Computing* üblich ist.

Gerade bei dem in naher Zukunft sowohl im horizontalen als auch im vertikalen Markt stark an Bedeutung gewinnenden Zugriff auf Informationen sind durch mobilitätsbasierte Implikationen neue Aspekte bei der Realisierung entsprechender Anwendungen zu berücksichtigen. Die Erforschung neuer Methoden in der Softwaretechnik zur Unterstützung der Softwareentwicklung ist dabei unumgänglich.

Evolution in der Datenbanktechnologie

Arnd G. Grosse

1 Einleitung

Die Integration rechnergestützter Werkzeuge, wie z.B. PPS-Systeme bzw. CAx-Systeme im ingenieurwissenschaftlichen Bereich, benötigen eine hohe informationstechnische Unterstützung, welches sich zum einen in der Forderung nach einem möglichst flexiblen Datenaustausch als auch zum anderen nach der Möglichkeit zur persistenten Datensicherung äußert. Die Datenbanktechnik stellt aus heutiger Sicht das geeignete Medium dar, um mit dieser Problematik fertig zu werden. Die heutige Verwendung von Datenbanken zur Modellierung von Anwendungsobjekten basierend auf dem relationalen Datenmodell ist mit einigen gravierenden Nachteilen verbunden, welches deren Verwendung im ingenieurwissenschaftlichen Bereich sehr umständlich macht bzw. fast ausschließt [KKK⁺94]:

- Die relationale Modellierung von Objekten verursacht durch die Segmentierung der Objekte auf verschiedene Relationen eine hohe Redundanz, wie auch ihre Wiedergewinnung über künstliche Schlüsselattribute. Hiermit verbunden sind wesentliche Verarbeitungsprobleme und signifikante Effizienzverluste.
- Die einzelnen Objekte ingenieurwissenschaftlicher Anwendungen weisen oft ein anwendungsspezifisches Verhalten auf, welches sich durch ein relationales Schema nicht erfaßt werden kann.

In den letzten Jahren hat nun ein Paradigmenwechsel stattgefunden hin zu objektorientierten Datenbanksystemen, welche versuchen, die genannten Nachteile zu vermeiden. Die Datenbanksysteme lehnen sich dabei sehr stark an objektorientierte Programmiersprachen, wie z.B. C++, an. Ein Objekt wird hierbei als die Verbindung von Struktur und Verhalten verstanden. Objekte innerhalb der Datenbank sind im Gegensatz zu relationalen Datenbanken, in welchen die Referenzierung mit Hilfe von Tupelattributswerten erfolgt, über Objektidentifikatoren, sog. OIDs, eindeutig referenzierbar. Hierbei lassen sich zwei Ausprägungen von OIDs unterscheiden:

- Zum einen kann es sich um physische bzw. physikalische OID handeln, welche die Position der Objektinstanz innerhalb der Datenbank repräsentieren. Dem Vorteil der leichten Auffindbarkeit von Objektinstanzen steht jedoch der Nachteil der aufwendigen Rekonfiguration der Datenbasis, z.B. durch Schemaänderungen, gegenüber.
- Die andere Möglichkeit besteht in der Verwendung logischer OIDs. Diese erfordern jedoch die Verwaltung der OIDs über interne Tabellen, welche die Zuordnung zur physikalischen Lokation übernehmen. Im Unterschied zu physikalischen OIDs ist jedoch eine Verschiebung der Instanzen ohne Beeinträchtigung der Referenz leicht möglich.

Im folgenden Abschnitt soll ein objektorientiertes Datenbanksystem, ObjectStore, exemplarisch vorgestellt und dessen Verwendung innerhalb des Sonderforschungsbereichs 346: „Rechnerintegrierte Produktion und Fertigung von Bauteilen“ beschrieben werden.

2 ObjectStore

Bei ObjectStore handelt es sich um ein Datenbanksystem, welches physische OIDs zur Referenzierung von Objektinstanzen verwendet. Die nachfolgende Abbildung 16 zeigt die heterogene Client-Server-basierte Architektur von ObjectStore [Obj93a]. Persistente Daten können hierbei auf verschiedenen Rechnern unter verschiedenen Servern gespeichert werden. Man spricht in diesem Kontext dann von einer Multiple-Client/Multiple-Server-Architektur.

Die Ausführung der Objektmethoden erfolgt unter ObjectStore immer auf Client-Seite. Eine wünschenswerte Ausführung der Methoden auf Server-Seite und die Rückübertragung der Berechnungsergebnisse ist jedoch nicht möglich.

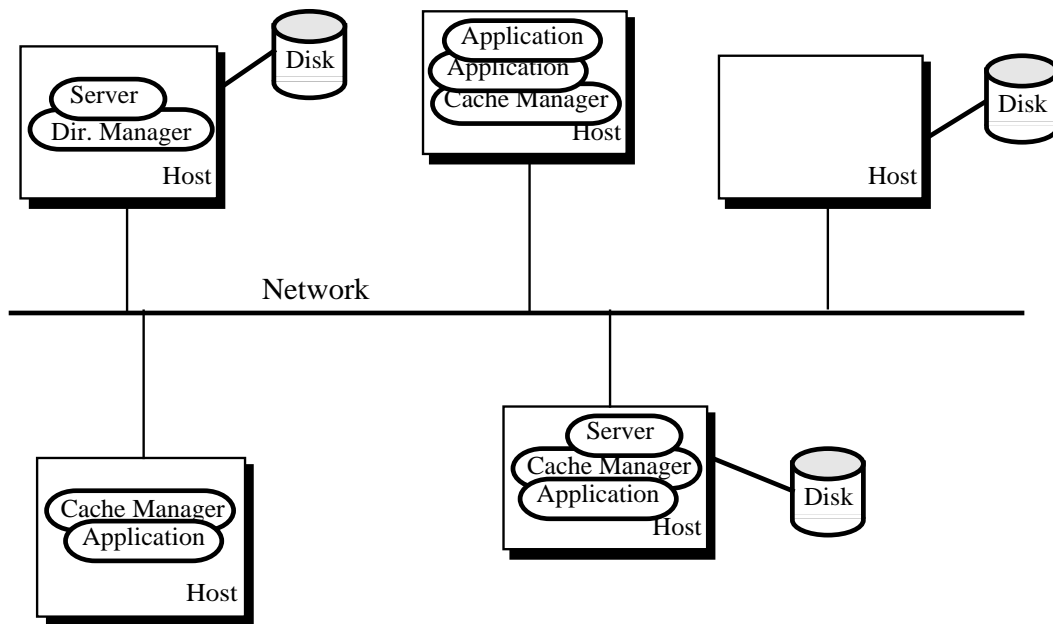


Abbildung 16. Architektur von ObjectStore

2.1 Architektur

ObjectStore erlaubt die Verwendung mehrerer Datenbank-Server, wobei jeder Server seine eigene Datenbank verwaltet und die Zugriffe darauf kontrolliert. Des weiteren übernehmen die Server Aufgaben der Lizenzverwaltung. Die Architektur von ObjectStore erfordert zudem, daß auf jedem Rechner auf dem ein Server läuft sich auch die physikalische Lokation der zu verwaltenden Datenbank befindet.

Jede ObjectStore-Anwendung (ObjectStore-Client) kann nur indirekt auf den ObjectStore-Server zugreifen. Auf jedem Rechnerknoten, auf dem eine Anwendung läuft, wird zum Zugriff auf den Server automatisch mit dem Start einer Anwendung ein Cache-Manager als Dämon-Prozeß installiert, der zum einen die Funktion besitzt, mittels Caching von Datenbank-Seiten, Zugriffe auf die Datenbank zu beschleunigen, als auch durch Verwaltung von Sperren eine Zugriffskontrolle auf den Objekten durchzuführen. Jede Anwendung greift mit Hilfe des Cache-Managers auf persistente Daten des Servers zu.

Der Cache-Manager enthält entweder die benötigten Informationen oder er erfragt sie seitenweise vom Server, wobei er für das Sperrenmanagement auf den jeweiligen Objekten verantwortlich ist. Sperren werden seitenweise und nicht objektweise vergeben, was unter ungünstigen Umständen zu Phantomsperrern führen kann. Bei Phantomsperrern handelt es sich um Sperren, welche bei objektweiser Betrachtung zu keiner Sperre führen dürften, jedoch entstehen, falls zwei zu sperrende Objekte sich auf der gleichen Speicherseite (Segment) in der Datenbank befinden. Werden mehrere Anwendungen auf einem Host gestartet, so übernimmt der zuvor gestartete Manager die genannten Aufgaben, wobei für jede Anwendung ein eigener Cache-Bereich verwaltet wird. Es existiert also maximal ein Cache-Manager pro Rechnerknoten.

Zusätzlich erlaubt ObjectStore die optionale Verwaltung für UNIX-Anwendungen transparenter Verzeichnisstrukturen. Hierzu dient einer oder mehrere Directory-Manager-Prozesse, wobei jedoch pro Server nur ein Directory-Manager existiert. Auf die vom Directory-Manager verwalteten Verzeichnisse läßt sich mit Hilfe an UNIX-Syntax angelehnter Hilfsprogramme zugreifen. So kann z.B. ein Verzeichnis mit Hilfe des Befehls `osmkdir <Verzeichnis>` angelegt bzw. dessen Inhalt mittels `osls <Verzeichnis>` angezeigt werden. Ebenso wie unter UNIX lassen sich hierarchische Verzeichnisstrukturen anlegen, wobei die Struktur logischer Natur ist. Mehrere Directory-Manager-Datenbasen können in verschiedenen File-Systemen gespeichert werden, bilden jedoch für die Anwendung eine logisch hierarchische Sicht. Einem ObjectStore-Verzeichnis ist somit keine physikalische Lokation zuzuordnen. Ebenso wie der Cache-Manager ist der Directory-Manager als Dämon-Prozeß realisiert.

2.2 Datenspeicherung und Zugriff

Der Zugriff auf persistente Daten erfolgt für die Anwendung transparent. Die Philosophie beim Entwurf von ObjectStore bestand darin, eine Gleichbehandlung zwischen transienten und persistenten Datenob-

jekten zu schaffen. Erreicht wird dies durch die Verwendung der Virtual Memory Mapping Architecture (VMMA). Sie besitzt die folgenden Eigenschaften [Obj93b]:

- VMMA garantiert, daß persistente Daten, welche durch ObjectStore gespeichert werden, in identischer Weise von C und C++-Programmen gehandhabt werden, wie dies bei transienten Daten der Fall ist. ObjectStore bezeichnet diese Eigenschaft als Persistenzunabhängigkeit des Datentyps.
- Die Verwendung dieser Technologie ermöglicht die Verwaltung von Daten mit minimalen Overhead, indem auf das virtuelle Speichermanagement des Betriebssystems zugegriffen wird. Diese Technik wird als virtuelle Speicherzuordnung bezeichnet.

Die Verwendung dieser Technik hat den Vorteil, daß es für Anwendungen transparent ist, um welche Art (transient bzw. persistent) von Objekt es sich handelt. Greift eine Anwendung auf ein Objekt zu, welches sich nicht im Hauptspeicher befindet, so verursacht dieser Zugriff einen Page-Fault. ObjectStore fängt diese Zugriffsschutzverletzung ab und schreibt für die Anwendung transparent das benötigte Datenobjekt aus der ObjectStore-Datenbasis in den virtuellen Speicher an diejenige Stelle, an der es von der Anwendung erwartet wird, und von wo es für Anwendungen dereferenzierbar ist. Nachfolgende Dereferenzierungen dieses Zeigers führen dann zu gewöhnlichen Zugriffen auf den virtuellen Speicher. Der große Vorteil dieser Methode liegt in der erreichbaren Geschwindigkeit bei Zugriffen auf persistente Daten, welche sich im günstigsten Fall nicht von Zugriffen auf transiente Objekte unterscheiden lassen.

2.3 Datenbankschema

Zur Verwendung persistenter Daten in Anwendungen ist im Rahmen von ObjectStore die Generierung von Schemainformationen notwendig. Hierbei handelt es sich um die Zusammenfassung von Datenstrukturen und funktionalen Schnittstellen. Bei objektorientierten Datenbanken sind dies die Klassenhierarchie, die Beschreibung der Methodenschnittstellen sowie die Attribute der Klassen. Zur Generierung von Schemainformationen ist in ObjectStore ein zweistufiger Prozeß nötig.

In einem ersten Schritt wird eine vorbereitende Schemadatenbasis erzeugt, die sog. Compilation Schema Database. Sie enthält Informationen über die Klassen innerhalb des Anwendungsquellcodes, die in einem persistenten Kontext verwendet werden, d.h. Klassen, deren Instanzen von oder in den persistenten Speicher geschrieben werden. Die Datenbasis wird dabei aus einer Schemadatei erzeugt, die der Autor der Anwendung bereitzustellen hat.

In einem weiteren Schritt, bevor die Anwendung gelinkt wird, produziert ein Schemagenerator die Anwendungsschemadatenbasis (Application Schema Database), die die Informationen über alle verwendeten internen und von Bibliotheken hinzugebundenen Klassen enthält und an das Anwendungsprogramm hinzugebunden werden muß. Diese Vorgehensweise birgt den Nachteil, daß zum Übersetzungszeitpunkt bereits bekannt sein muß, welche Objekttypen innerhalb der Anwendung persistent verwendet werden. Eine nachträgliche Erweiterung um neue Datentypen, erfordert die Neukompilation der Anwendung.

3 Visionäre Verwendung der Datenbanktechnologie

Wie im vorangegangenen Abschnitt aufgezeigt wurde, ist die Struktur der objektorientierten Datenbank durch ein relativ starres Client/Server-Schema geprägt. Diese Konzept ist im lokalen bzw. kleinen Maßstab gut handhabbar. Jedoch mit zunehmender Größe und stärkerer Vernetzung von Unternehmensbereichen, wie sie durch die Einführung von CIM-Technologien und PPS entstehen, ist die Verwendung dieser Datenbanktechnologie unzureichend. Hierzu bedarf es neuer Mechanismen und Methoden, um eine bessere Kooperation und Effizienz zu erzielen. Lösungen hierfür wurden bereits im Rahmen des SFB 346 „Rechnerintegrierte Produktion und Fertigung von Bauteilen“ erarbeitet. So sind dies:

- Die Erweiterung des Begriffes des Objektes um Aktivität mit eigenem Kontrollfluß, wodurch eine Instanz der Produktionswelt leichter zu modellieren ist [KKK⁺94].
- Die Unterstützung von Objektmigration. Objekte können hierzu flexibel auf unterschiedliche Rechner verteilt werden und dabei gleichzeitig über geeignete Kommunikationsmechanismen im Rahmen einer gemeinsamen Anwendung kooperieren [KKK⁺94].
- Die lose Kopplung von Datenbanken durch das Cluster-Konzepts und die Wissenspropagierung von Objektinstanzen mit Hilfe von Trading. Hierdurch läßt sich eine dynamische Adaption des Gesamtsystems realisieren [KK94a]. Zudem vermeidet diese Lösung die ineffiziente Realisierung über eine physikalisch zentrale Datenbank.

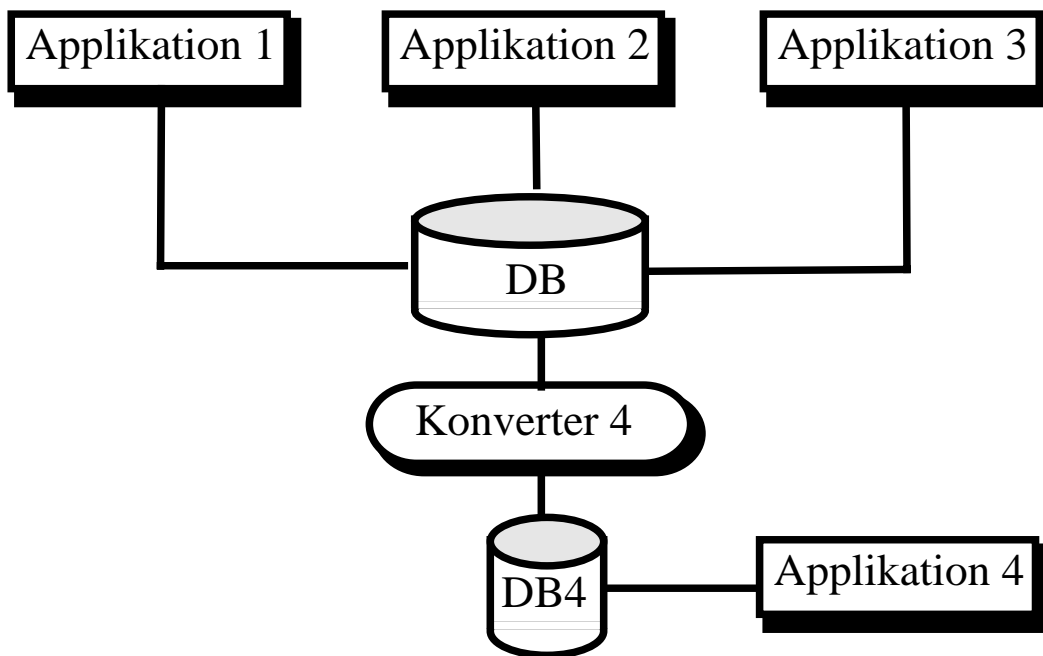


Abbildung 17. Integration vorhandener und zukünftiger Datenbankapplikationen

Für die Anwendungen wird hierdurch eine logisch zentrale Sicht der Datenbank realisiert, wie sie in Abbildung 17 gezeigt wird. Damit auch ältere Applikationen an einem solchen System teilnehmen können, ist die Ankopplung von älteren Datenbanken über Konverter vorzusehen.

Wie bereits im vorangegangenen Kapitel erläutert wurde, erlaubt ObjectStore die Ausführung von Objektmethoden nur auf Client-Seite. Dieses Vorgehen kann unter Umständen sehr aufwendig werden, wenn zur Berechnung eine Vielzahl von großen Objektinstanzen durch das Datenbanksystem zu übertragen ist. Im Rahmen weiterer Arbeiten des SFB 346 ist zu untersuchen, inwieweit eine solche Berechnung auf Server-Seite erkannt und durchgeführt werden kann.

4 Zusammenfassung

Wie anhand der objektorientierten Datenbank ObjectStore erläutert wurde, besitzen objektorientierte Datenbanken im Vergleich zu relationalen Datenbanken eine wesentliche größere Funktionalität bei der Modellierung einer komplexen Miniwelt. Jedoch stellen moderne ingenieurwissenschaftliche Anwendungen hohe Anforderungen bezüglich der Kooperation und Kommunikation an die heutigen Objektmodelle, welches nur unzureichend durch Datenbanken, wie z.B. ObjectStore, unterstützt wird.

Wie in einer Umfrage gezeigt wurde [Ker94], ist in Zukunft eine höhere Kooperation zwischen unterschiedlichen Unternehmensbereichen erforderlich. Dieses muß durch geeignete informationstechnische Mechanismen unterstützt werden. Als Ausgangsbasis bieten sich hierfür objektorientierte Datenbanken an. Jedoch ist hierzu eine Reihe weiterer Arbeiten notwendig, um eine logisch zentrale Datenbanksicht bei physikalischer Verteilung zu erzielen, um hierdurch die geforderte Kooperation zu erreichen. Hierzu werden derzeit eine Reihe von Konzepten im Rahmen des Sonderforschungsbereichs 346 „Rechnerintegrierte Produktion und Fertigung von Bauteilen“ erforscht.

Kooperation autonomer Agenten

Jörn Hartroth

1 Software-Agenten

Der Begriff des Software-Agenten ist eines der aktuell besonders häufig verwendeten Schlagworte in der Informatik und stellt den Gegenstand intensiver Aktivität in ganz unterschiedlichen Forschungsbereichen dar. Ansätze kommen besonders aus Richtung der (verteilten) künstlichen Intelligenz, der Robotik, der computerunterstützten Arbeit und der (verteilten) objektorientierten Programmierung. Das rege Interesse an der Beschäftigung mit Agenten richtet sich dabei in verschiedenen Disziplinen auf ganz unterschiedliche Aspekte, was dazu führt, daß die jeweils entworfenen Agentenmodelle in ihren Eigenschaften wenig miteinander gemein haben und der Agentenbegriff eine etwas unscharfe Gestalt annimmt ([WJ95]). Eine gewisse Eingkeit besteht darüber, daß ein Agent eine *aktive* Instanz darstellt, die ihre im jeweiligen Szenario definierte Umwelt — die eventuell auch andere Agenten umfaßt — beobachtet, mit ihr interagiert und selbständig auf die Lösung einer vorgegebenen Aufgabe hinarbeitet. Zur Veranschaulichung der verschiedenen Aspekte agentenorientierter Systeme stellt Abbildung 18 eine funktionale Untergliederung des Agenten in die für lokale Berechnungen zuständige Rumpfkomponte und den für die Verbindung zur Außenwelt zuständigen Kopf vor.

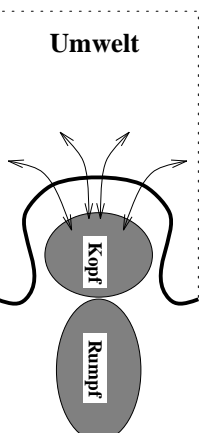


Abbildung 18. Funktionale Unterteilung eines Softwareagenten

Grundsätzlich sind innerhalb der verschiedenen Agentenmodelle die zwei Ausprägungen des Einzelagentensystems (*single agent system*, *SAS*) und des Multiagentensystems (*multi agent system*, *MAS*) zu unterscheiden. Einzelagentensysteme beschäftigen sich vordringlich mit lokalen Abläufen innerhalb eines Agenten und seiner Wechselwirkung mit der Umwelt, Multiagentensysteme legen die Betonung auf die Wechselbeziehungen zwischen mehreren Agenten und das emergente Gesamtverhalten von Agentengruppen.

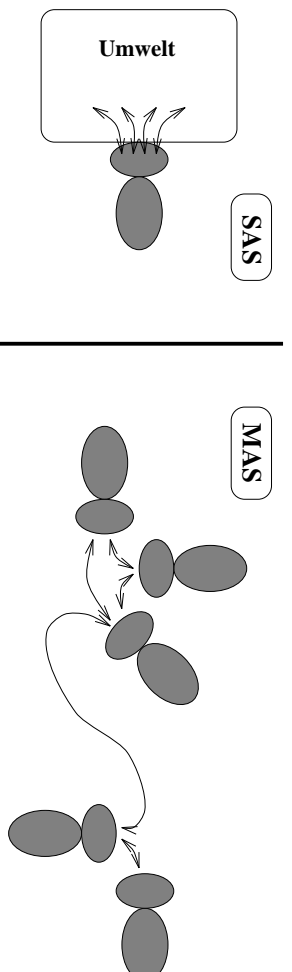


Abbildung 19. Einzelagent und Multiagentensystem

2 Unterschiedliche Agentenmodelle

Im folgenden Abschnitt sollen die von den einzelnen Forschungsgebieten betrachteten Aspekte von Agentensystemen näher vorgestellt werden.

2.1 Künstliche Intelligenz

Das Gebiet der künstlichen Intelligenz (*KI*) verfolgt seit den siebziger Jahren das Ziel, basierend auf einem symbolischen Verarbeitungsmodell² für die kognitiven Abläufe im menschlichen Gehirn geeignete Verfahren zur maschinellen Problemlösung zu entwickeln. Wesentliche Punkte bilden die Aquisition und symbolische Repräsentation von Wissen über unterschiedliche Weltmodelle und die automatische Generierung von Plänen zur Erfüllung vorgegebener Ziele. Als (autonomer) Agent wird eine Instanz angesehen, die lokales Wissen über ihre Umwelt enthält, selbsttätig Pläne generiert und verfolgt und die auf beobachtete Veränderungen in der Umwelt oder geänderte Ziele eigenständig durch dynamische Planänderung reagieren kann. Die Umwelt liegt bei den klassischen KI-Ansätzen außerhalb des kognitiven Systems, so daß nach der eingangs gewählten Unterscheidung Einzelagentensysteme vorliegen, bei denen der Schwerpunkt auf den lokalen Abläufen im Agentenrumpf liegt.

2.2 Verteilte künstliche Intelligenz

Seit dem Ende der achtziger Jahre entwickelt sich im Bereich der KI zunehmendes Interesse an der Untersuchung der Problemlösung durch kooperatives Zusammenwirken von Instanzengruppen. Die Betrachtung erweitert sich unter der Bezeichnung "verteilte künstliche Intelligenz" von der *mikroskopischen* Sicht der Abläufe in einer einzelnen Instanz zur *makroskopischen* Sicht auf die Interaktion zwischen den Gruppenmitgliedern und das resultierende Gesamtverhalten ganzer Populationen kognitiver Instanzen — mithin von Multiagentensystemen. Ursprünglich ausgehend von einer zentralistischen Erstellung geteilter Pläne tritt in neuerer Zeit der Gedanke der Autonomie einzelner Agenten verstärkt in der Vordergrund: Wissen über die Umwelt existiert nicht zentral sondern verteilt über die Agenten und wird unter ihnen durch Nachrichten ausgetauscht. Jeder Agent plant unabhängig und allein basierend auf lokalem Wissen sein weiteres Vorgehen. Innerhalb dieses Rahmens ergeben sich durch unterschiedliche Wahl einiger zentraler Grundannahmen eine Reihe von Modellen mit recht verschiedenen Eigenschaften. Die Wahlmöglichkeiten umfassen:

- einheitliche oder unterschiedliche (z.B. unvollständige) Weltsicht zwischen den Agenten,
- identische oder unterschiedliche Fähigkeiten der Agenten,
- ein gemeinsames Ziel für eine Agentenpopulation oder jeweils eigene — evtl. kollidierende — Ziele einzelner Agenten,
- prinzipielle Uneigennützigkeit der Agenten, Angebot von Diensten gegen Entlohnung oder Steuerung durch (hierarchische) Befehlsstrukturen.

Die Kombination von identischen Agenten mit gemeinsamem Ziel und uneigennützigem Verhalten führt zum Modell des *verteilten Problemlösens*, unterschiedlich befähigte Agenten mit eigenen Zielen und Dienstkonzept auf Entlohnungsbasis ergeben einen offenen Agentenmarkt. Als theoretische Basis für die Modellierung dieser Agenteneigenschaften und einer Reihe von Kooperationsformen dienen Ergebnisse aus der Soziologie ([GB92], Seite 5), mithin wird eine Nachahmung der menschlichen Interaktionsformen in bestimmten Szenarien angestrebt. Zuverlässige Kommunikation zwischen Agenten wird in der VKI als gegeben postuliert, in der Regel in Form nachrichtenbasierter Mechanismen oder eines Blackboard ([CL92]). Großes Gewicht wird jedoch angesichts der gewollten Diskrepanz zwischen dem verfügbaren Wissen unterschiedlicher Agenten auf die Semantik der Kommunikationsakte gelegt. So verwenden die Ansätze nach [Sho90] und [Bou91] eine Klassifizierung von Nachrichten nach ihrer *Intention*, abgeleitet von Untersuchungen über menschliche Sprachakte ([Sea69]). Mit *KQML* (der *knowledge query and manipulation language*) sind Standardisierungsbemühungen auf dem Gebiet des Wissensaustausches im Gange ([FWW⁺93]). Die betrachteten Systeme sind demnach Multiagentensysteme mit Betonung auf der Funktionalität der Agentenrumpfe, in denen auch die Interpretation von Kommunikationsakten stattfindet.

2.3 Robotik

Die Robotik verbindet die in der klassischen und verteilten KI entwickelten Modelle mit Realisierungen von in der realen Welt arbeitenden Sensoren und Aktuatoren. Aufgaben von Agenten richten sich in der Regel auf die Fortbewegung des Roboters selbst oder die Bewegung von Gegenständen durch Manipulatoren. Die Robotik liefert der künstlichen Intelligenz realitätsnahe Szenarien und steht in enger Verbindung mit den in KI und VKI erzielten Ergebnissen.

² im Gegensatz zum konnektionistischen Modell der *Neuronalen Netze*

2.4 Computerunterstütztes kooperatives Arbeiten und Mensch–Maschine Kooperation

Das Anliegen der Mensch–Maschine Kooperation besteht darin, menschliche Benutzer durch den Einsatz von Computern von stereotypen Tätigkeiten, wie z.B. Such- und Sortiervorgängen, möglichst weitgehend zu entlasten. Durch computerunterstütztes kooperatives Arbeiten (*computer supported cooperative work, CSCW*) sollen ihm darüberhinaus die im persönlichen Umgang gewohnten Interaktionsformen mit anderen Menschen geboten werden, wobei zusätzlich eine Entkopplung des räumlichen und zeitlichen Zusammenhangs zwischen den Kooperationspartnern erreicht und die Nutzung komfortabler Werkzeuge ermöglicht werden soll.

Einer der Kooperationspartner ist in jedem Fall der Mensch, Agenten treten entweder in der Rolle eines Vermittlers zwischen Mensch und Maschine auf (auch als *Interface-Agent* bezeichnet, etwa in Form eines Mail-Agenten, der automatisch eingehende Sendungen nach den Vorlieben des Benutzers sortiert, [Mae94]) oder übernehmen die Rolle eines Kooperationspartners (z.B. bei der Anmeldung von Besuchern in einem Labor, siehe [KSCK93]). Es handelt sich hier in der Regel um Einzelagenten, bei denen sowohl Kopf- als auch Rumpffunktionalität betrachtet wird.

2.5 Verteilte objektorientierte Systeme

Die verteilte objektorientierte Programmierung (auch *object based concurrent programming, OBCP*, siehe [GB92]) nähert sich dem Agentenbegriff aus Richtung der Beschäftigung mit programmiersprachlichen Konzepten unter starker Berücksichtigung des Umgangs mit Daten und Kontrollabläufen. Agenten stellen in diesem Umfeld die Erweiterung der bereits bekannten Modelle der Datenmobilität (Message Passing, Parameterübergabe beim RPC) und Codemobilität (Objektmigration) um die zusätzliche Mobilität von Aktivitäten dar. Beim Vorgang der Agentenmigration werden somit alle drei für den programmiersprachlichen Ansatz wesentlichen Aspekte gemeinsam einem Ortswechsel unterworfen. Die bei diesem Agentenbegriff zu lösenden Aufgaben umfassen Erweiterungen der typischen Fragen der verteilten Programmierung, etwa

- Steuerung der Mobilität, Identifikation und Auffindung beweglicher Instanzen,
- Notwendigkeit für spezielle Protokolle (z.B. Replikation, Persistenz) zur Einhaltung der geforderten Konsistenzkriterien in Gegenwart fehlerträchtiger Kommunikationswege,
- Behandlung von Heterogenitäten bei verschiedenen Hardware- oder Softwareplattformen im System,
- Schutz gegen Einblicknahme oder absichtliche Verfälschung von übertragenen Daten.

Zum Teil können bestehende Ansätze aus dem Bereich der konventionellen verteilten Programmierung übernommen werden, im Hinblick auf die mit dem Aufkommen des Agentenbegriffs einhergehende Erweiterung der Dynamik und Komplexität vorgeschlagener Systeme und die zunehmende Dezentralisierung auch organisatorischer Konzepte werden jedoch völlig neue Lösungen erforderlich. Hier spielt die Funktionalität der Agentenköpfe in einem Multiagentensystem eine wesentliche Rolle.

3 Aufgaben der Kooperation

Die besondere Eigenschaft des relativ jungen Agentenmodells besteht darin, daß gegenüber früheren Ansätzen in der KI und der (verteilten) objektorientierten Programmierung das vorgestellte Gesamtsystem nicht länger einer konzeptionell zentralisierten Kontrolle unterliegt. Expertensysteme, Wissensbasen oder herkömmliche verteilte Datenbanken oder Anwendungen — wie komplex auch in ihrer internen Struktur — wurden mit Blick auf klar definierte Aufgaben entworfen und von einem gewünschten Verhalten abgeleitet. Der Ansatz der Multiagentensysteme verfolgt die Vorstellung, einzelne Bausteine in Form autonomer Instanzen zu definieren, die jeweils nur einen kleinen Ausschnitt des Gesamtsystems überblicken und deren Verhalten von dieser eingeschränkten Sicht bestimmt ist, so daß bei geeignetem Zusammenwirken ein emergentes Verhalten entsteht, das dem gewünschten Systemverhalten möglichst nahe kommen soll³. Diese Sichtweise spiegelt sich z.B. in so visionären Vorstellungen wie dem offenen elektronischen Dienstemarkt wieder, zu dessen Eigenschaften hohe Dynamizität und Autonomie auch in Bereichen der Kontrollfunktionen zählen.

Zentraler Punkt der Multiagentensysteme ist die Fähigkeit der einzeln konstruierten und je nach Anlage des Szenarios in ihren Attributen ganz unterschiedlichen Agenten, in sinnvoller Weise zu interagieren. Das Zusammenwirken von Agenten zur Erreichung jeweils gesteckter Ziele soll unabhängig

³ Die Abgrenzung zu *Neuronalen Netzen* besteht darin, daß Agenten auf einer höheren konzeptionellen Ebene eigenständig (begrenzte) Aufgaben erfüllen können und mit einem wesentlich komplexeren Verhaltensrepertoire ausgestattet sind als Neuronen.

von den bereits in Abschnitt 2.2 aufgezählten unterschiedlichen Modalitäten als *Kooperation* bezeichnet werden. Dieser Kooperationsbegriff ist vom in der Mensch-Maschine-Kooperation und dem CSCW verwendeten gleichlautenden Term zu unterscheiden, da dort der Mensch mit seinen sozialen und kommunikativen Fähigkeiten die zentrale Rolle einnimmt. Im folgenden sollen die in der KI untersuchten kognitiven Implikationen der Agenteneigenschaft nicht weiter betrachtet werden.

Das Mittel der Interaktion zwischen Agenten besteht im gewählten Umfeld im Austausch von Nachrichten. Jedem Agenten steht ein Vorrat von Nachrichten verschiedenen Inhalts zur Verfügung, die er aussenden kann und auf deren Empfang er reagieren kann⁴. Das hier betrachtete Modell geht davon aus, daß der Ablauf des Nachrichtenaustausches im Rahmen einer Kooperation strukturiert erfolgt, indem die aktuell für einen legalen Kooperationsablauf möglichen Nachrichten abhängig von der Historie zurückliegender Interaktionen eingeschränkt werden. Die Art der Einschränkung des Nachrichtenraums ist spezifisch für die Art der Kooperation, sie bildet einen den beteiligten Agenten zugeordneten *Kooperationskontext*. Der Kooperationskontext für den einfachen Fall einer Kaufbeziehung mag etwa in einer Abfolge $\{\text{Auftragserteilung} \rightarrow \text{Warenübergabe} \rightarrow \text{Bezahlung}\}$ bestehen, vereinfacht dargestellt in Abbildung 20. Es ist zu erwarten, daß autonome Agenten zur Erfüllung ihrer Aufträge dynamisch Kooperationsbeziehungen eingehen und wieder lösen, wobei nicht selten der Fall auftreten wird, daß Agenten gleichzeitig Teilnehmer an mehreren Kooperationskontexten sind. Im allgemeinen Fall ist abzusehen, daß dabei gegenseitige Abhängigkeiten zwischen Kooperationskontexten auftreten, die den Nachrichtenraum gegenüber dem isoliert betrachteten Kontext erweitern oder weiter einschränken.

Zur Veranschaulichung dieser Sicht soll ein vereinfachtes Beispiel aus dem Handel dienen. Ein einfacher Kooperationskontext *Kaufvertrag* (Abbildung 20 links) beschreibt die Abwicklung eines Warenverkaufs in Form der bereits oben genannten Nachrichtenfolge. Bis zur erfolgten Bezahlung bleibt die Vertragsbeziehung zwischen Käufer und Verkäufer bestehen, und der Verkäufer behält sich das Eigentum an der Ware vor. Der Käufer aus diesem einfachen Szenario kann wiederum als Wiederverkäufer auftreten und über einen weiteren Kooperationskontext in Beziehung mit dem Endkunden stehen, dem er die Ware ausliefert, bevor sie beim ursprünglichen Verkäufer bezahlt ist (Abbildung 20 rechts). Offensichtlich besteht eine Beziehung zwischen den beiden dargestellten Kooperationskontexten, die sich im realen Vorbild in einem sog. *verlängerten Eigentumsvorbehalt* des ursprünglichen Verkäufers gegenüber dem Endkäufer äußert — bei Ausfall des Wiederverkäufers würde der resultierende Kooperationskontext etwa eine Bezahlungsnachricht vom Endkunden an den Verkäufer vorsehen.

Gegenstand der weiteren Forschung wird es sein, Einflüsse zwischen unterschiedlichen Kooperationskontexten explizit zu erfassen und dynamisch entstehende übergreifende Effekte behandeln zu können, die beim Entwurf der einzelnen Kooperationskontexte nicht vorgesehen werden konnten.

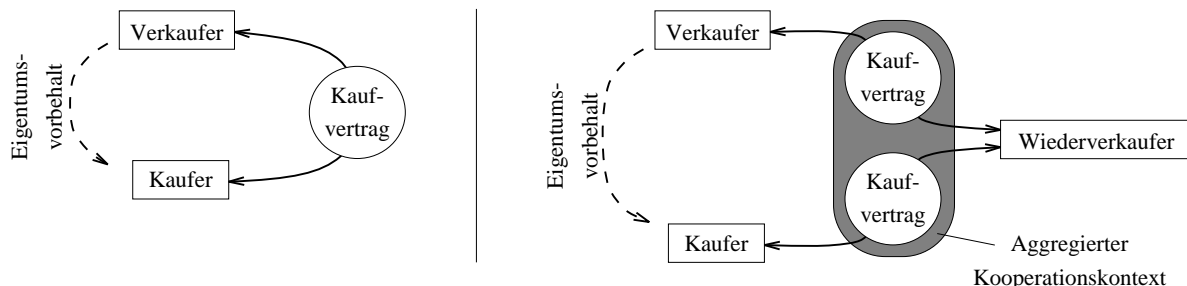


Abbildung 20. Beziehung zwischen Kooperationskontexten — Beispiel des verlängerten Eigentumsvorbehalts

4 Ziele

Im weiteren Verlauf sollen geeignete Spezifikationsmethoden für Kooperationskontexte entwickelt werden, die es ermöglichen, dynamisch auftretende und wieder unwirksam werdende Wechselbeziehungen zwischen unabhängig definierten Kontexten zu erfassen und zur automatischen Generierung von resultierenden Kooperationskontexten zu verwenden. Ergänzend dazu wird eine konforme Umsetzungsmethodik benötigt, die den Nachrichtenablaufs gemäß den Vorgaben des jeweils gültigen Kontextes regelt.

⁴ Grundsätzlich besteht an dieser Stelle das Problem, Syntax (wegen möglicher Heterogenitäten) und Semantik der auszutauschenden Nachrichten zu behandeln. Da Heterogenitätsbehandlung bereits in herkömmlichen verteilten Systemen betrachtet wird und Modellierung von Semantik im Bereich der VKI intensiv erforscht wird, sollen diese Aspekte hier nicht weiter untersucht werden.

Dienstgüteunterstützung für Gruppenkommunikation

Markus Hofmann

1 Einleitung

Verteilte multimediale Anwendungen erfordern von zugrundeliegenden Kommunikationsdiensten oftmals die Einhaltung bestimmter Qualitätsmerkmale. Dies betrifft sowohl quantitative Kriterien, wie beispielsweise Verzögerung und Durchsatz, als auch qualitative Merkmale, wie etwa die Zuverlässigkeit der Verbindung. Soll eine ausgehandelte Dienstgüte für die Dauer der Kommunikation garantiert werden, so müssen innerhalb des Netzwerkes die benötigten Ressourcen vor der eigentlichen Datenübertragung reserviert werden. Da Betriebsmittel nur in beschränktem Umfang zur Verfügung stehen, führt eine überdimensionierte Reservierung oder die unbedachte Nutzung reservierter Ressourcen zu einer hohen Ablehnungsquote weiterer Verbindungsaufbau-Wünsche. Insbesondere bedarf es im Zusammenhang mit dem Datenaustausch innerhalb einer Gruppe von Benutzern besonderer Überlegungen hinsichtlich des Umgangs mit Betriebsmitteln. Abhängigkeiten zwischen den Gruppenmitgliedern können zur effizienten Nutzung der Ressourcen herangezogen werden. Darüberhinaus ermöglichen neuartige Protokollmechanismen eine deutliche Reduzierung der durch Multicast-Verbindungen verbrauchten Ressourcenmenge.

Im Rahmen dieses Artikels werden Probleme bei der Ressourcenreservierung für Mehrpunkt-Verbindungen herausgearbeitet. Einfache Beispielszenarien verdeutlichen die Problematik. Ebenso werden mögliche Lösungsansätze diskutiert. Zum Abschluß wird ein neuartiges Konzept zur Unterstützung von zuverlässigen Multicast-Verbindungen vorgestellt, mit dessen Hilfe die Menge der benötigten Betriebsmittel innerhalb globaler heterogener Netzwerke deutlich reduziert werden kann.

2 Ressourcenreservierung für Mehrpunkt-Verbindungen

Die Ressourcenreservierung für Multicast-Verbindungen unterscheidet sich in mehreren Punkten von derjenigen für klassische Unicast-Verbindungen. Zum einen muß bei der Aushandlung der Dienstgüte ein gemeinsamer Konsens zwischen den beteiligten Kommunikationsteilnehmern gefunden werden. Zum anderen erlauben anwendungsspezifische Abhängigkeiten eine gemeinsame Nutzung von Ressourcen durch mehrere Gruppenmitglieder.

2.1 Aushandlung der Dienstgüte

Die meisten Reservierungsprotokolle [SZ95d] zeichnen sich durch ein verbindungsorientiertes Konzept aus, wobei während der Verbindungsaufbauphase eine Datenstruktur mit der angeforderten Dienstqualität (Flow Specification, FlowSpec) an alle involvierten Systeme verteilt wird. Im Rahmen eines 2- oder 3-Wege-Handshakes wird schließlich die von allen Kommunikationssystemen unterstützte Dienstgüte ausgehandelt und die entsprechenden Ressourcen reserviert.

Soll im Falle einer Gruppenkommunikation eine global einheitliche Dienstqualität ausgehandelt werden, so müssen zu deren Festlegung die Anforderungen der einzelnen Empfänger bekannt sein. Erst nachdem alle Kommunikationsteilnehmer dem Sender die von ihnen unterstützte bzw. geforderte Dienstgüte mitgeteilt haben, kann dieser aus den Parametern der Empfänger die globalen Ende-zu-Ende-Werte ermitteln. Beispielsweise errechnet sich die maximale Ende-zu-Ende-Verzögerung aus dem Maximum der Verzögerungen zu den einzelnen Empfängern hin, während der Übertragungsabschnitt mit minimalem Durchsatz den minimalen Gesamtdurchsatz bestimmt. Sind für einzelne Werte sowohl obere als auch untere Schranken angegeben, kann es bei der Aushandlung zu Konflikten kommen. Fordert ein Empfänger E_x beispielsweise einen minimalen Durchsatz von x Mbps und unterstützt ein weiterer Empfänger E_y lediglich einen Durchsatz von y Mbps mit $y < x$, so kann für die gesamte Empfängergruppe keine global einheitliche Dienstgüte bestimmt werden. In diesem Fall muß ein Algorithmus zur Konfliktauflösung angestoßen werden, der beispielsweise einen der beiden Empfänger aus der Kommunikationsgruppe ausschließt. Eine formalisierte Darstellung der skizzierten Problematik findet sich in [MB94].

2.2 Data Filtering

Eine feinere Skalierung der Dienstgüte ermöglicht das Filtern von Benutzerdaten innerhalb der Zwischensysteme. Dadurch muß nicht mehr allen Empfängern der Multicast-Verbindung eine global einheitliche

Dienstgüte garantiert werden. Vielmehr wird diese individuell für jeden Kommunikationsteilnehmer entsprechend seiner Anforderungen bestimmt und die Ressourcen auf den jeweiligen Übertragungswegen gemäß der getroffenen Vereinbarung reserviert. Als Beispiel für ein solches Szenario diene Abbildung 21. Das sendende Endsystem C speist einen farbigen Videofilm in das Netz ein. Station B kann jedoch nur Schwarz-Weiß-Videos darstellen, wohingegen der Benutzer des Endsystems A den Film in Farbe betrachten möchte. Im Falle einer global gültigen Dienstgarantie müssen auf allen Verbindungsstrecken die

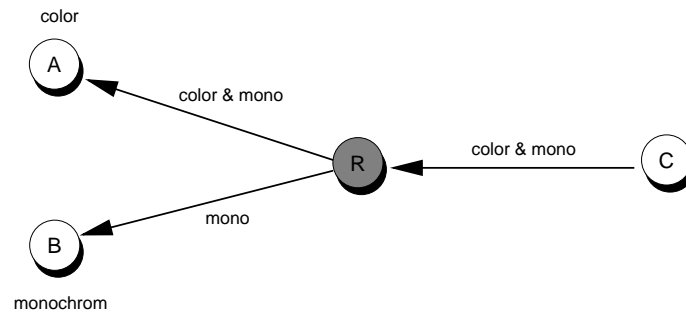


Abbildung 21. Filtern der Benutzerdaten in Zwischensystemen

Ressourcen zur Übertragung der Film- und Farbinformation reserviert werden. Wird zur Übertragung des Videos eine Codierung verwendet, welche die Trennung von Film- und Farbinformation erlaubt, kann das Zwischensystem R die Farbinformationen aus dem Datenfluß herausfiltern und lediglich die Filminformation an das Endsystem B weiterleiten. Mit der zu übertragenden Datenmenge ändern sich auch die Werte verschiedener Dienstgüteparameter. So ist zur Übertragung des Schwarz-Weiß-Videos auf der Strecke zwischen R und B beispielsweise ein geringerer Durchsatz erforderlich als auf der Verbindung von R nach A zur Übermittlung des Farb-Videos. Damit können für die Datenübertragung zwischen R und B auch weniger Ressourcen reserviert werden. Diese Vorgehensweise hilft zum einen Konflikte bei der Aushandlung der zu garantierenden Dienstgüte zu umgehen. Zum anderen werden auf den einzelnen Übertragungsabschnitten nur diejenigen Ressourcen reserviert, die zur Erbringung des gewünschten Dienstes auch wirklich benötigt werden. Die Granularität der Skalierung ist jedoch nicht beliebig fein wählbar. Sie hängt von der Codierung der zu übertragenden Daten ab. Diese muß das Herausfiltern bestimmter Informationen erlauben. In obigem Beispiel wurde dies durch die getrennte Übertragung der Film- und Farb-Information ermöglicht. Eine detaillierte Darstellung der Mechanismen zum Filtern von Anwendungsdaten ist in [WHD94] zu finden.

2.3 Resource Sharing

Treten während des Datenaustauschs innerhalb einer Gruppe von Benutzern mehrere Endsysteme als Sender auf, so werden oftmals verschiedene 1:n-Verbindungen aufgebaut und der Datenverkehr über diese abgewickelt. Jeder Sender unterhält eine eigene Multicast-Verbindung hin zu den restlichen Gruppenmitgliedern. Diese Verbindungen werden als voneinander unabhängig behandelt und stehen unter dieser Betrachtungsweise in keinem Zusammenhang zueinander. Deshalb ist die explizite Reservierung von Ressourcen für jede einzelne der Verbindungen notwendig. In vielen Fällen resultiert diese Vorgehensweise in einer Überbelegung der Betriebsmittel, was an Hand des in Abbildung 22 dargestellten Szenarios erläutert werden soll. Zwischen den Benutzern der Endsysteme A, B und C soll eine Audio-Konferenz abgehalten werden. Hierzu baut jedes der drei Endsysteme eine Multicast-Verbindung zu den beiden anderen hin auf. Aus Gründen der Übersichtlichkeit wurden in der Abbildung lediglich die Multicast-Verbindungen der Endsysteme A und B dargestellt. Im Rahmen einer Audiokonferenz ist normalerweise zu einem beliebigen Zeitpunkt nur ein einzelner Sprecher aktiv. Mehrere Personen sprechen nur dann gleichzeitig, wenn sie sich um die Zuteilung des Wortes bemühen. Eine solche Situation erstreckt jedoch über einen kleinen Zeitraum hinweg, während dessen eine Verminderung der Übertragungsqualität durchaus akzeptabel ist.

Von Interesse ist in diesem Szenario die Übertragungsstrecke zwischen dem Vermittlungssystem R und dem Endsystem C. Ist eine gemeinsame Nutzung von Betriebsmitteln ausgeschlossen, werden im Netz für die beiden Multicast-Verbindungen unabhängige Reservierungen vorgenommen. Da nach obigen Überlegungen zu den meisten Zeitpunkten jedoch nur ein Sprecher aktiv ist, werden zu den meisten Zeitpunkten auch nur über eine der aufgezeigten Verbindungen Daten übertragen. Demnach wird von

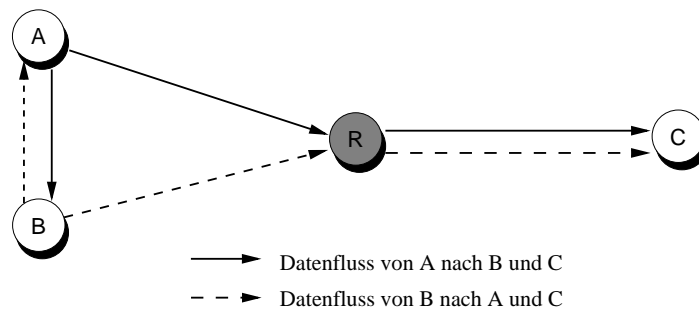


Abbildung 22. Gemeinsame Nutzung von Betriebsmitteln

den Ressourcen, die das Kommunikationssystem für die Audiokonferenz reserviert hat, zu jedem Zeitpunkt nur ein Bruchteil wirklich benötigt. Ist die gemeinsame Nutzung von Betriebsmitteln möglich, wird das Kommunikationssystem über die Abhängigkeit zwischen den Multicast-Verbindungen informiert. Da die Summe des Datenverkehrs auf den Multicast-Verbindungen zu den meisten Zeitpunkten demjenigen einer einzelnen Verbindung entspricht, wird das Kommunikationssystem auf der Übertragungsstrecke von R nach C auch nur die Ressourcen für eine Verbindung reservieren. Die belegten Betriebsmittel können jedoch von allen Multicast-Verbindungen, die zur Audiokonferenz gehören, genutzt werden. Während der Zeitspannen, in denen sich mehrere Kommunikationsteilnehmer um das Wort bemühen, sind die reservierten Betriebsmittel unter Umständen nicht ausreichend. Eine Verminderung der Übertragungsqualität in solchen Situationen ist jedoch akzeptabel. Zudem wird in manchen Konferenzsystemen das Senderecht durch ein übergeordnetes Protokoll geregelt. Im BERKOM-II Multimedia Collaboration Teleservice [MMC92] kann das Senderecht beispielsweise über Token gesteuert werden. Dadurch ist sichergestellt, daß zu jedem Zeitpunkt stets nur ein Sender aktiv ist.

Das obige Beispiel kann leicht auf Anwendungen ausgedehnt werden, bei denen zwar mehrere Sender gleichzeitig aktiv werden können, ihre Anzahl jedoch weiterhin unter der Summe aller potentieller Sender liegt. Zur Realisierung des Resource Sharings muß dem Benutzer des Kommunikationsdienstes die Möglichkeit gegeben werden, Abhängigkeiten zwischen einzelnen Verbindungen zu definieren. Eine Erweiterung der Dienstschnittstelle um entsprechende Parameter ist unabdingbar. Erste Vorschläge für eine solche Erweiterung und die Definition entsprechender Protokollmechanismen werden in [GHM94] gemacht.

3 Zuverlässige Gruppenkommunikation

Einige Anwendungen, wie beispielsweise verteiltes Rechnen oder die Realisierung eines verteilten virtuellen Speichers, erfordern vom zugrundeliegenden Kommunikationsdienst die Bereitstellung einer zuverlässigen Multicast-Verbindung. Soll diese basierend auf einem unzuverlässigen Netzwerkdienst realisiert werden, müssen Mechanismen zur Erkennung und Behebung von Fehlern in das Kommunikationsprotokoll integriert werden. Wie in [Hof94] gezeigt, kann durch die Wahl geeigneter Protokollmechanismen die Belastung der beteiligten Kommunikationsknoten und Verbindungsstrecken innerhalb globaler heterogener Netzwerke deutlich reduziert werden. Zu diesem Zweck wurde das *Konzept der Lokalen Gruppen* entwickelt.

3.1 Das Konzept der Lokalen Gruppen

Die grundlegende Idee des Konzepts der Lokalen Gruppen besteht in der Zusammenfassung nahe beieinander gelegener Kommunikationsteilnehmer zu lokalen (Unter-) Gruppen. Jede lokale Gruppe schließt genau ein ausgezeichnetes Endsystem, den *Lokalen Gruppenverwalter*, ein. Dieser wird nicht statisch festgelegt, sondern kann während der Lebenszeit der lokalen Gruppe dynamisch wechseln. Der Verwalter ist für die korrekte Übertragung der Daten an die Mitglieder der lokalen Gruppe verantwortlich und tritt gegenüber dem Multicast-Sender als ihr Repräsentant auf. Dem Verwalter ist die Identität aller Mitglieder der ihm zugeordneten lokalen Gruppe bekannt. Prinzipiell ist jedes Endsystem in der Lage, die Rolle eines Gruppenverwalters zu übernehmen. Somit ist der Aufbau und die Unterhaltung spezieller Kommunikationsknoten zur Verwaltung der Untergruppen nicht notwendig. Stattdessen erfolgt die Gruppierung der Kommunikationsteilnehmer und die Wahl geeigneter Gruppenverwalter dynamisch. Ein Beispiel für die Aufteilung einer globalen Kommunikationsgruppe in mehrere lokale Untergruppen ist in Abbildung 23 dargestellt.

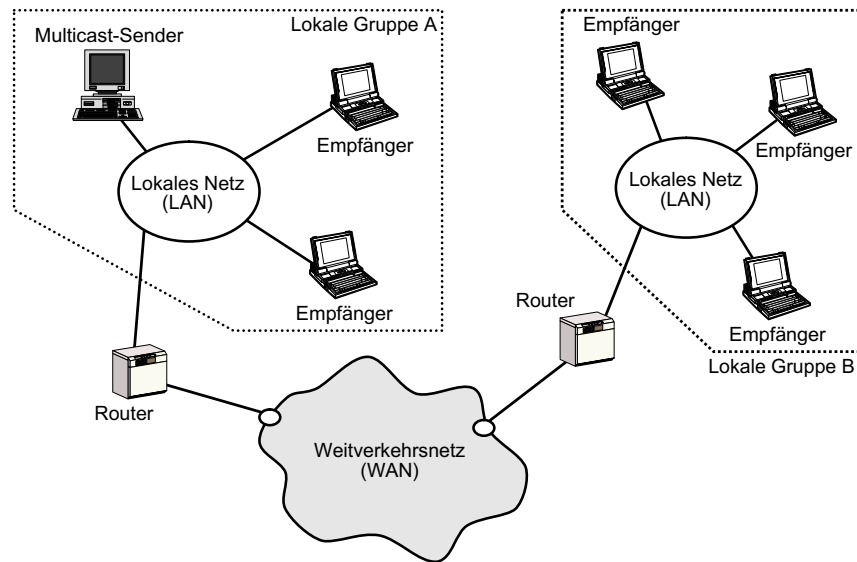


Abbildung 23. Aufteilung in lokale Untergruppen

3.2 Aufbau lokaler Gruppen und Datentransfer

Möchte ein Endsystem an der Gruppenkommunikation teilnehmen, so muß es sich zunächst einem geeigneten Gruppenverwalter und damit einer lokalen Gruppe zuordnen. Wird eine solche nicht gefunden, gründet das Endsystem eine neue lokale Gruppe und ernennt sich selbst zu deren Verwalter. Bei der Suche nach einem geeigneten Gruppenverwalter können unterschiedliche Metriken (z.B. Verzögerung, Durchsatz, Distanz,...) zugrundegelegt werden. Der Datentransfer gestaltet sich damit nach folgendem Schema: Der Sender übermittelt die Datenpakete unter Nutzung der Multicast-Verbindung an alle Gruppenmitglieder. In regelmäßigen Zeitabständen stellt der Sender Statusanfragen an alle Kommunikationsteilnehmer. Diese werden von allen Empfängern, mit Ausnahme der Gruppenverwalter, mit kurzer zufälliger Verzögerung beantwortet. Diese Verzögerung verhindert das zeitgleiche Senden von Quittungen durch mehrere Endsysteme, was in busbasierten Netzwerken zu Kollisionen führen würde. Die Statusmeldungen sind nicht an den Multicast-Sender gerichtet. Vielmehr werden diese als Unicast-Pakete an den jeweiligen Gruppenverwalter übermittelt. Hierfür wird bereits zu Beginn der Multicast-Kommunikation eine Unicast-Verbindung aufgebaut. Der Gruppenverwalter sammelt die eintreffenden Quittungen innerhalb einer vorgegebenen Zeitspanne und wertet diese unter Berücksichtigung seines eigenen Status aus. Dadurch erhält er Kenntnis über die korrekt empfangenen Datenpakete der einzelnen Mitglieder. Ihre Vereinigung ergibt die von der lokalen Gruppe korrekt empfangene Datenmenge. Der Gruppenverwalter fordert, falls möglich, die ihm fehlenden Daten bei den Mitgliedern seiner lokalen Gruppe über die Unicast-Verbindungen an. Nur diejenigen Datenpakete, die kein Mitglied der lokalen Gruppe korrekt empfangen hat, werden beim Sender oder bei einer benachbarten lokalen Gruppe geordert. Nachdem die Mitglieder der lokalen Gruppe die angeforderten Daten über Unicast-Verbindungen dem Verwalter übermittelt haben, führt dieser lokale Übertragungswiederholungen durch. Hierbei sendet er fehlerfrei empfangene Nachrichten an alle Gruppenmitglieder, welche die jeweiligen Datenpakete noch nicht korrekt erhalten haben.

Wie in [Hof94] gezeigt wurde, reduziert die Durchführung von Übertragungswiederholungen innerhalb der lokalen Gruppen deutlich die Belastung des Gesamtnetzes. Zudem wird durch die parallele Auswertung von Quittungen innerhalb der lokalen Gruppenverwalter der Multicast Sender stark entlastet, was zu einer Verringerung der Übertragungsverzögerung führt.

4 Ausblick

Zur Realisierung eines bedarfsgerechten und effizienten Kommunikationssystems zur Unterstützung von Gruppenkommunikation bedarf es neuartiger Konzepte und Methoden. Im Hinblick auf eine sparsame Reservierung und effiziente Nutzung von Netzwerkressourcen durch Multicast-Verbindungen müssen spezielle Mechanismen entwickelt und in die Kommunikationsprotokolle integriert werden. Begleitend hierzu gilt es die Neuentwicklungen an Hand geeigneter Simulationsmodelle oder realer Messungen kritisch zu bewerten.

Zuverlässige Dienste in offenen Systemen: Anforderungen und Konzepte

Ludwig Keller

1 Ausgangspunkt

Verteilte Systeme werden immer stärker als Medium zur Einbeziehung externer Dienstleistungen verstanden. Eine solche Unterstützung ist jedoch nur durch eine effiziente Kommunikationstechnik und durch geeignete infrastrukturelle Maßnahmen für verteilte Systeme möglich. Während eine effiziente Kommunikationstechnik durch den Einsatz der Glasfasertechnologie und der ATM-Protokolle möglich ist, existieren derzeit nur unzureichende adäquate infrastrukturelle Mechanismen, deren Einsatz die mit der Integration der Dienstleistungen in der lokalen Anwendung einhergehende Informationsflut beherrschbar machen.

Ein Ansatz zur infrastrukturellen Organisation verteilter Systeme ist ein dedizierter Akquisitionsmechanismus, der Trading-Dienst, der in verschiedenen Entwicklungsplattformen, wie z.B. AMP ANSAware und ISO ODP, und zahlreichen Forschungsprojekten zur Bewältigung der Informationsflut und zur Vermittlung einzelner Dienstleistungen zur Laufzeit herangezogen wird [Kel93]. Ein Trader, der diesen Dienst im System anbietet, nimmt hierbei Dienstangebote von autonomen Dienstobjekten entgegen und vermittelt diese unter Einbeziehung ihrer statischen und dynamischen Eigenschaften auf Anfrage einer Anwendung (Dienstbenutzer, Klient). Somit wird ein offener Dienstemarkt im verteilten System etabliert. Eine wichtige Diensteigenschaft in einem solchen Markt ist die Zuverlässigkeit, deren Berücksichtigung zukünftig an Bedeutung gewinnt.

So weisen verteilte Systeme trotz der Verbesserung der Kommunikationstechnologie eine hohe Fehleranfälligkeit ihrer Systemkomponenten auf, die für zahlreiche Anwendungen nicht akzeptabel ist. Es sind deswegen Mechanismen erforderlich, mit denen Dienstleistungen fehlertolerant erbracht werden können. Dies ist möglich, da durch die Vielzahl in einem verteilten System angebotenen funktional äquivalenten Dienstleistungen eine natürlich Redundanz von Dienstobjekten vorhanden ist, die zur fehlertoleranten Diensterbringung eingesetzt werden kann.

Dieser Beitrag stellt das System *Agora* vor, welches zur Vermittlung und Erbringung zuverlässiger Dienstleistungen in einem offenen System entwickelt wurde. Hierzu wird im folgenden Abschnitt das Dienstmodell von *Agora* erläutert. Anschließend werden Beschreibungsmöglichkeiten für die Zuverlässigkeit eines Dienstes behandelt. Im vierten Abschnitt wird dann das Systemmodell von *Agora* dargestellt.

2 Dienstmodell

Der Gegenstand der Vermittlung eines Traders sind Dienstleistungen (kurz Dienst). Ein *Dienst* umfaßt eine Menge von Dienstmethoden, die von einem Objekt oder einer Menge kooperierender Dienstobjekte erbracht werden. Er definiert sich somit durch das an einer Dienstschnittstelle für eine Anwendung sichtbare Verhalten. Jeder Dienst wird systemweit eindeutig durch seinen *Diensttyp* identifiziert. Dieser legt den *Schnittstellentyp*, d.i. die Menge der Methodensignaturen, und eine Liste von *Diensteigenschaftstypen* fest. Diese Eigenschaftstypen können abhängig von ihrer Änderungsrate statisch oder dynamisch sein. Statische Diensteigenschaften können beim Trader mit dem Dienstangebot lokal verwaltet werden, während dynamische Eigenschaften zum Zeitpunkt der Dienstvermittlung von den Dienstobjekten direkt erfragt werden müssen.

Dienste können sowohl von einem Dienstobjekt oder durch eine Menge kooperierender Dienstobjekte erbracht werden. Der Dienst eines Dienstobjektes wird als *Basisdienst* bezeichnet, der Dienst von mehreren kooperierenden Dienstobjekten als *zusammengesetzter Dienst*. Zusammengesetzte Dienste, deren Dienstobjekte Dienste verschiedener Diensttypen anbieten, werden als *verteilte Dienste* bezeichnet. Stellen die Dienstobjekte hingegen Dienste vom gleichen Diensttyp bereit, kann diese Redundanz zur Vermittlung *zuverlässiger Dienste* genutzt werden. Fällt mit der Diensterbringung ein Dienstobjekt aus, kann durch geeignete Rekonfigurationsmaßnahmen dieser Ausfall dadurch toleriert werden, daß ein anderes Dienstobjekt, das den gleichen Dienst anbietet, anstelle des ausgefallenen Dienstobjektes den benötigten Dienst erbringt.

Ein zusammengesetzter Dienst läßt sich beschreiben durch eine Menge von *Dienstobjektgruppen*. Die Dienstobjekte einer Dienstobjektgruppe sind zum einen durch ihren gemeinsamen Diensttyp und zum anderen durch ihre (unterschiedliche) Lokation im verteilten System bestimmt.

3 Zuverlässigkeitsgarantien

Das Bereitstellen und die Benutzung mehrerer Dienstobjekte zur Erbringung zuverlässiger Dienste in einem offenen verteilten System verursacht für den Dienstbenutzer Kosten. Um zwischen Kosten und Zuverlässigkeit eines Dienstes die für die Anwendung benötigte Dienstobjektauswahl treffen zu können, ist die Beschreibung der Zuverlässigkeit unumgänglich.

Zur Beschreibung der Zuverlässigkeit werden *Zuverlässigkeitskenngrößen* \mathcal{F} eingesetzt, die basierend auf dem zugrundeliegenden verteilten System \mathcal{G} die Zuverlässigkeit eines zusammengesetzten Dienstes \mathcal{Z} quantifizieren: $\mathcal{F}(\mathcal{G}, \mathcal{Z}) \rightarrow \mathcal{W}$ mit \mathcal{W} als Wertebereich. Sie müssen in einem offenen verteilten System, in dem verschiedene Hersteller, Organisationen und Teilnehmer involviert sind, folgende Anforderungen erfüllen:

- *Berücksichtigung von Redundanz*: Ein zusammengesetzter Dienst kann aus Dienstobjekten bestehen, die Dienste eines gemeinsamen Diensttypes bereitstellen. Die durch die Redundanz erhöhte Zuverlässigkeit des Dienstes muß durch die Kenngröße zum Ausdruck kommen.
- *Berücksichtigung von Dienstobjektgruppen*: Die Beschreibung muß auch Dienstobjekte verschiedener Dienstobjektgruppen unterscheiden, d.h., daß Dienstobjekte verschiedener Diensttypen involviert sind.
- *Hohe Verteilungstransparenz*: Den Aspekt, daß die zu vermittelnden Dienste verteilt erbracht werden, insbesondere die Ausprägung des verteilten Systems soll durch die Zuverlässigkeitskenngröße so weit wie möglich verborgen werden.
- *Hohe Aussagekraft*: Die Zuverlässigkeitsbeschreibung soll einer Anwendung nachvollziehbar sein. Hierzu muß die Beschreibung eine möglichst hohe Aussagekraft über das Ausfallverhalten des Dienstes widerspiegeln.
- *Polynomiale Berechnungszeit*: Da sich der Handlungsbereich eines Traders über einen größeren Bereich eines verteilten Systems erstrecken kann und die Vermittlung von Diensten angemessene Bearbeitungszeiten des Traders erfordert, ist es notwendig, daß die Berechnung der Zuverlässigkeit höchstens polynomial erfolgt.
- *Beschränkung auf öffentliche Informationen*: Die vom Trader zu vermittelnden Dienstobjekte können sich in Subnetzwerken befinden, die von verschiedenen Organisationen betreut werden. Dies bedingt, daß der Trader zur Zuverlässigkeitsbeschreibung nur auf Informationen aufsetzen kann, die öffentlich verfügbar sind. Diese beschränken sich in offenen Systemen auf die Netzwerktopologie und die Lokation der Dienstobjekte.

In der Literatur wurden bereits zahlreiche Zuverlässigkeitskenngrößen entwickelt, wie z.B. die Verfügbarkeitswahrscheinlichkeit (Reliability), die s-t-Konnektivität und die Knoten-Konnektivität [Shi91]. Ihnen ist gemein, daß sie die oben genannten Anforderungen nicht alle erfüllen. So lassen sich, mit Ausnahme der Verfügbarkeitswahrscheinlichkeit, weder Redundanzen von Dienstobjekten noch die Komposition eines Dienstes als Kollektiv von mehreren Dienstobjektgruppen beschreiben. Beides ist mit der Verfügbarkeitswahrscheinlichkeit möglich; diese setzt jedoch voraus, daß zur Zuverlässigkeitsberechnung die Verfügbarkeitswahrscheinlichkeiten der Netzwerkkomponenten bekannt sind. Zudem ist diese Kenngröße NP-hart [Bal86].

Insbesondere die Anforderung nach polynomialer Berechnungszeit und die Einschränkung auf öffentliche Informationen schränken die Möglichkeit der Zuverlässigkeitsbeschreibung sehr stark ein. Für das System *Agora* wurden deswegen für ein zugrundeliegendes verteiltes System \mathcal{G} und einem zu betrachtenden zusammengesetzten Dienst \mathcal{Z} folgende vier neuen Zuverlässigkeitskenngrößen entwickelt:

- *Gruppenoperabilität* $\tilde{\kappa}(\mathcal{G}, \mathcal{Z})$: berechnet die minimale Anzahl an Netzwerkkomponenten aus \mathcal{G} , die zwei Dienstobjektgruppen des zusammengesetzten Dienstes \mathcal{Z} voneinander trennt, so daß kein Dienstobjekt einer Dienstobjektgruppe mit einem Dienstobjekt der anderen Dienstobjektgruppe miteinander kommunizieren kann. Fallen zur Dienstleistung im ungünstigsten Fall diese Netzwerkkomponenten aus, kann der vermittelte Dienst nicht erbracht werden.
- *Stabilität der Konnektivität* $\sigma_{\tilde{\kappa}}(\mathcal{G}, \mathcal{Z})$: berechnet die Anzahl an Einzelkomponentenausfälle $E_{\mathcal{Z}}$, die die Gruppenoperabilität eines zusammengesetzten Dienstes \mathcal{Z} im Netzwerk \mathcal{G} verschlechtert. Sie ist definiert als $\sigma_{\tilde{\kappa}}(\mathcal{G}, \mathcal{Z}) = 1/(E_{\mathcal{Z}} + 1)$. Die Stabilität ist stets im Zusammenhang mit der Gruppenoperabilität zu untersuchen und gibt Aufschluß über die Beeinflussbarkeit der Gruppenoperabilität.
- *Toleranzrate* $\tau(\mathcal{G}, \mathcal{Z})$: berechnet sich aus dem Minimum der S,T-Toleranzrate zweier Dienstobjektgruppen S,T aus \mathcal{Z} . Diese ist definiert als das Verhältnis der Gruppenoperabilität von S,T zu deren kürzesten Distanz. Je kürzer diese Distanz ist, umso weniger Komponenten können das Scheitern der Dienstleistung bewirken.

- *RK-Verhältnis* $\vartheta(\mathcal{G}, \mathcal{Z})$: berechnet sich aus dem Verhältnis der minimalen Redundanz einer Dienstobjektgruppe aus \mathcal{Z} zu der Anzahl der minimal benötigten Netzwerkkomponenten, um sämtliche Dienstobjekte miteinander zu verbinden. Je mehr Netzwerkkomponenten zur Dienstleistung involviert sind, umso unzuverlässiger ist der Dienst und umso größer ist das RK-Verhältnis.

In Tabelle 1 werden die Kenngrößen von *Agora* hinsichtlich der o.a. Anforderungen bewertet. Zudem wurde die Verfügbarkeitswahrscheinlichkeit in der Tabelle mit aufgenommen.

Anforderung	P	$\tilde{\kappa}(\mathcal{G}, \mathcal{Z})$	$\tilde{\kappa}(\mathcal{G}, \mathcal{Z}) \wedge \sigma_{\tilde{\kappa}}(\mathcal{G}, \mathcal{Z})$	$\tau(\mathcal{G}, \mathcal{Z})$	$\vartheta(\mathcal{G}, \mathcal{Z})$
Redundanz	++	++	++	+	+
Objektgruppen	++	+	+	+	+
Verteilungstransparenz	++	+	+	o	o
Detaillierungsgrad	o	+	++	++	++
Polynomial	–	+	+	+	+
Information	–	+	+	+	+

Tabelle 1. Beurteilung der *Agora*-Zuverlässigkeitskenngrößen

Ein kleines Anwendungsbeispiel sollen den Gebrauch der neu entwickelten Zuverlässigkeitskenngrößen dokumentieren. In Abbildung 24 ist ein verteiltes System dargestellt, in dem ein zusammengesetzter Dienst „*DVI_Print*“ bereitgestellt wird. Dieser Dienst setzt sich aus zwei Basisdiensttypen zusammen, nämlich einen DVI-Postscript-Übersetzer *dvips* und einen Druckdienst *lpr*. Jedes der Diensttypen bildet eine Dienstobjektgruppe. In Tabelle 25 wird die im Beispiel erreichte Zuverlässigkeit berechnet. Die Gruppenoperabilität ist durch die Redundanz der *dvips*-Dienstobjekte eingeschränkt. Durch die beiden Bustopologien (M1,M5) kommt die geringe Konnektivitätsstabilität zustande; der Ausfall von R1,M1,R2,R7,M5 oder R10 reduziert die Gruppenoperabilität auf 1. Durch die kurze Distanz der Dienstobjekte bei den Bustopologien wird eine relativ hohe Toleranzrate erreicht. Sie besagt, daß aufgrund der minimalen Distanz der Dienstobjektgruppen prinzipiell ein zuverlässiger Dienst möglich ist. Das RK-Verhältnis hingegen gibt durch seinen relativ geringen Wert Auskunft, daß sehr viele Netzwerkkomponenten an der Dienstleistung involviert sind. Dies bedeutet, daß die zu berücksichtigenden Dienstobjekte weit im verteilten System verstreut sind und deswegen insgesamt den Dienst unzuverlässig machen. In Abb. 24 sind die zur Verbindung sämtlicher Dienstobjekte erforderlichen Netzwerkkomponenten durch dicke Linien hervorgehoben.

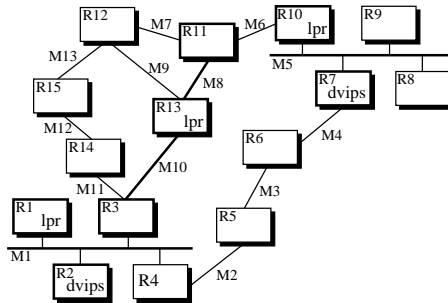


Abbildung 24. Verteiltes System

Redundanz(<i>dvips</i>):	2
Redundanz(<i>lpr</i>):	3
Redundanz(<i>DVI_Print</i>):	$\min(2,3) = 2$
Gruppenoperabilität:	2
Stabilität:	$\frac{1}{6}$
Kürzeste Distanz:	3
Toleranzrate:	$\frac{2}{3}$
RK-Verhältnis:	$\frac{2}{12}$

Abbildung 25. Kenngrößen

4 Systemmodell

Das um zusammengesetzte Dienste erweiterte Dienstmodell wurde im System *Agora* realisiert. Die wichtigsten Anforderungen an ein solches System, welches die Vermittlung und die Erbringung zuverlässiger und verteilter Dienste gestattet, sind:

- *Entkopplung der Konzepte Dienst und Dienstobjekt*: Der Dienstbenutzer soll von dem eigentlichen Dienstobjektaufruf entkoppelt sein und lediglich den Dienst als abstrakte Instanziierung der Dienstausführung aufrufen. Diese Anforderung ist notwendig, um die Gleichwertigkeit von Diensten mehrerer Dienstobjekte behandeln zu können.
- *Kompositionstransparenz*: Dem Dienstbenutzer soll die Anzahl der zur Erbringung eines Dienstes involvierten Dienstobjekte, d.h. dessen innere komplexe Dienststruktur verborgen sein. Ein Dienstbenutzer kann somit nicht zwischen einem Basisdienst und einem zusammengesetzten Dienst unterscheiden.

- *Server-Transparenz*: Dem Dienstbenutzer soll die Identität der Dienstobjekte verborgen bleiben. Als *Server* wird in *Agora* der Netzwerkknoten bezeichnet, auf dem das Dienstobjekt allokiert ist.
- *Fehlertransparenz*: Dem Dienstbenutzer soll der Ausfall von Dienstobjekten bei der Erbringung eines zuverlässigen Dienstes im Rahmen des Fehlermodells verborgen sein. Das Systemmodell muß somit die Voraussetzung schaffen, geeignete Fehlertoleranzprotokolle zu betreiben, um den Ausfall von Dienstobjekten während der Dienstleistung maskieren zu können.

Als Randbedingung liegt dem System die *Fail-Stop-Fehlersemantik* zugrunde.

Beim herkömmlichen Trading-Ansatz interagieren Dienstbenutzer und Dienstobjekt stets direkt. Soll dem Dienstbenutzer ein Dienst fehlertolerant erbracht werden, ist die Entkopplung von Dienstbenutzer und Dienstobjekten unumgänglich. Diese wird durch eine zusätzliche Instanz, dem *Dienstadministrator*, ermöglicht. Diese bildet eine transparente Dienstschnittstelle zwischen dem Dienstbenutzer und den Dienstobjekten und erlaubt somit die Entkopplung der Konzepte Dienst und Dienstobjekt. Der genaue

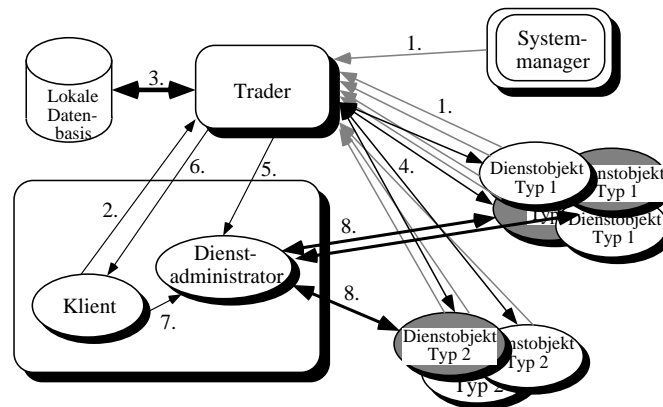


Abbildung 26. Systemmodell vom System *Agora*

Ablauf eines Dienstauftrages in *Agora* ist in Abb. 26 beschrieben. Dienstobjekte können wie in herkömmlichen Trading-Systemen Dienstangebote an den Trader exportieren (1). Zudem kann der Systemmanager über eine interaktive Schnittstelle beim Trader zusammengesetzte Dienste anmelden. Hierzu müssen vom Systemmanager u.a. die benötigten Basisdiensttypen, die Abbildung der Diensteigenschaften des zusammengesetzten Dienstes auf die der Basisdienste sowie eine Dienststruktur spezifiziert werden, die die Aufrufreihenfolge zur Erbringung des Dienstes festlegt [Kel94]. Nach einer Dienstanfrage des Klienten (2) überprüft der Trader zunächst, ob es sich um einen Basisdienst oder einen zusammengesetzten Dienst handelt. Mittels seiner lokalen Datenbasis (3) und dem direkten Anfragen der potentiellen Dienstobjekte (4) ermittelt der Trader sodann die Dienstobjekte, die zur Erbringung des Dienstes eingesetzt werden sollen. Anschließend initialisiert er den Dienstadministrator auf dem Rechner des Klienten mit den Adressen der involvierten Dienstobjekte und der Dienststruktur (5) und übergibt abschließend die Adresse des Dienstadministrators dem Klienten (6). Mittels dieser ruft der Klient den Dienst beim Dienstadministrator auf (7) und stößt somit die Dienstausführung an (8).

Die zum Export eines zusammengesetzten Dienstes notwendige Bindeinformation wird in [Kel94] erläutert. Auf die formale Beschreibungssprache zur Spezifikation eines zusammengesetzten Dienstes, auf die Erbringung des Dienstes im Dienstadministrator sowie die Berücksichtigung von geeigneten Fehlertoleranzprotokollen wird in [KG95] genauer eingegangen. Zudem werden darin einige Realisierungsaspekte von *Agora* erläutert.

5 Zusammenfassung

Dieser Beitrag stellte das System *Agora* zur Vermittlung und Erbringung zuverlässiger und zusammengesetzter Dienste in offenen verteilten Systemen vor. Wichtige Voraussetzung zur Vermittlung zuverlässiger Dienste ist die Beschreibungsmöglichkeit ihrer Zuverlässigkeit. Hierzu wurden vier neue Zuverlässigkeitskenngrößen, nämlich die Gruppenoperabilität, die Konnektivitätsstabilität, die Toleranzrate sowie das RK-Verhältnis, eingeführt, die sich durch die zur Berechnung der Kenngrößen benötigten eingeschränkten Informationen über das System und durch ihre polynomiale Berechnungszeit auszeichnen.

Abgekoppelte Objekte im Mobile Computing

Dietmar Kottmann

1 Einleitung

Durch die zunehmende Miniaturisierung von Rechnerkomponenten und die Entwicklung leistungsfähiger Mobilkommunikationssysteme wurde in den letzten Jahren das Gebiet „Mobile Computing“ für breite Anwendungsfelder erschlossen. Nachdem die Entwicklung elementarer Systemkomponenten, wie Funknetze oder stromsparender Displays, einen gewissen Reifegrad erreicht hat, steht heute die Frage der Anwendungsunterstützung durch höhere Systemmechanismen im Vordergrund.

Konzepte zum Entwurf und zur Unterstützung von Anwendungen in verteilten Systemen bedürfen wesentlicher Änderungen, falls mobile Teilnehmer mit integriert werden sollen [KG94]. Neue Anforderungen ergeben sich dadurch, daß (teil-)autonomes Arbeiten den Randbedingungen der asymmetrischen Ressourcenausstattung und der beschränkten Konnektivität im Mobile Computing am besten gerecht wird. Zentrale Fragestellung ist dabei immer wieder die (lokale) Unterstützung von Zugriffen auf gemeinsam benutzte Datenobjekte [Kot95b].

2 Lösungsalternativen

Durch Verwendung dedizierter Caching- oder Replikationsalgorithmen werden Zugriffe auf gemeinsam benutzte Daten unter Einschränkung der Abstimmung und Kommunikation mit anderen Rechnern möglich. Für die technische Umsetzung ergeben sich dabei die folgenden Alternativen:

Klassische Replikation: Diese Replikationsalgorithmen stammen üblicherweise aus dem Datenbank- oder Fehlertoleranzkontext und garantieren Ein-Kopien-Serialisierbarkeit als Korrektheitskriterium. Partitionierungen, wie sie im Mobile Computing unvermeidlich sind, galten beim Entwurf der Algorithmen als extreme Ausnahmefälle. Deshalb beruhen die Algorithmen auf einer engen Synchronisation der einzelnen Kopien, die sich im Kontext des Mobile Computing verbietet.

Gruppenkommunikationsprotokolle als Basis der gemeinsamen Arbeit: Replikation von Datenobjekten beruht generell auf der Auslieferung von Zugriffen auf mehreren Kopien. Um allein durch Auslieferungsgarantien ohne weitergehende Mechanismen für viele Fälle hinreichende Synchronitätsbedingungen bieten zu können, ist mindestens eine totale und kausale Auslieferung nötig [CS93]. Trotzdem ist der Ansatz nicht als allgemeines Paradigma geeignet. Zum einen läßt sich nicht jede gewünschte Synchronität allein durch Auslieferungsgarantien erreichen [CS93], zum anderen kann die notwendige Totalität nur durch Verzögern des Ausliefern von Nachrichten bis zur Zustimmung aller Kopien erreicht werden, was sich im Mobile Computing wiederum verbietet.

Replikation mit schwachen Konsistenzgarantien: Eine Abschwächung der Garantien für die Anwendung ist für Spezialfälle eine hinreichende Lösung. Allerdings resultiert auch dies in keinem allgemeingültigen Verfahren [Gol92].

Read-Only Replikation: Read-Only Replikation kann als Lösung für die Fälle dienen, in denen ausschließlich Lesezugriffe auf gemeinsam benutzte Daten benötigt werden. Die naheliegende Erweiterung, Schreibzugriffe zu puffern und sie bei einer Zentralkopie auszuliefern, ist nicht hinreichend, da die Schreiboperationen in der Warteschlange bei späteren Lesezugriffen keine Berücksichtigung finden — aus Sicht des Benutzers also bis zum Kopienabgleich verschwunden sind.

Auch das Puffern und zusätzliche lokale Durchführen der Operationen ist keine befriedigende Lösung. Ein Beispiel für dabei mögliche Anomalien ist in [Kot95a] zu finden, in dem ein Beispielablauf präsentiert wird, für den es keinen äquivalenten Ein-Kopien-Ablauf gibt. Ein derartiges Vorgehen wäre folglich nicht replikationstransparent.

Optimistische Replikation: Bei optimistischer Replikation dürfen auf allen Objekten des Systems alle Operationen durchgeführt werden. Werden später Konflikte festgestellt, so wird mindestens eine der Operationen zurückgenommen. Folglich ist der Mechanismus in Reinform nur in den Bereichen anwendbar, in denen die Konfliktwahrscheinlichkeit vernachlässigbar ist.

Verschiedene Erweiterungen existieren im Hinblick auf die Konfliktbehandlung. Die Reinform, daß Operationen ausgewählt und gelöscht werden, ist in vielen Fällen um die Alternative, Kopien zu sperren und die Auflösung auf manuellen Eingriff zu delegieren, erweitert. Dabei spricht man von einem Reintegrationslauf.

Als Fazit kann gezogen werden, daß es zwar für Spezialfälle befriedigende Lösungsalternativen gibt, doch keiner der Vorschläge als generische Lösung hinreichend ist.

Das im Bereich des Mobile Computing fortgeschrittenste System ist das verteilte Dateisystem CODA [KS92], das im wesentlichen auf der Idee der optimistischen Replikation beruht. In CODA wurde dafür der Begriff der *abgekoppelten Operation* geprägt. Wie aber Erfahrungen mit dem Einsatz zeigten, ist dieses Paradigma in seiner Grundform nur für Objekte, die selten gemeinsam benutzt werden, hinreichend [SKM⁺93]. Gerade für diese ist aber ein Unterstützungskonzept wesentlich. Dazu kann CODA allein dienen, da es als Dateisystem keine Information über die interne Struktur der von ihm verwalteten Daten besitzt und somit nur in den seltensten Fällen ausreichend Informationen für eine automatische Konfliktauflösung anbieten kann.

In aktuellen Arbeiten wird basierend auf der Grundidee der optimistischen Replikation ein System realisiert, das *abgekoppelte Objekte* ermöglicht. Das System verfolgt das Ziel, das angestrebte (*teil-* *autonome*) Arbeiten zu ermöglichen. Dabei kann das hier vorgestellte System durch Ausnutzen von Wissen über die Semantik der grundlegenden Objekte über bekannte Ansätze hinausgehen. Für Einzelheiten sei auf [Kot93, Kot95a] verwiesen. Im folgenden wird ein kurzer Überblick über wichtige Aspekte vermittelt, wobei die spezielle Frage im Vordergrund steht, wie die risikoreiche optimistische Vorgehensweise in vielen Fällen durch Ausnutzen der Objektsemantik vermieden werden kann und dennoch für viele Anwendungsfälle der Eindruck uneingeschränkter Verfügbarkeit erbracht wird.

3 Abgekoppelte Objekte

Die verfolgte Vorgehensweise beruht auf zwei grundlegenden Erweiterungen. Zum einen ist dies das konsequente Ausnutzen der nach außen sichtbaren Semantik von Objekten, zum anderen die Erkenntnis, daß in einem verteilten System keine einheitliche Zeitbasis existiert und folglich gewisse Anwendungsteile in der Vergangenheit oder auch Zukunft ablaufen können, ohne daß es für einen Benutzer sichtbar ist. Auch Zeitsprünge können zusätzlich als Mittel eingesetzt werden, um in vielen Fällen ohne Optimismus korrekte Systemläufe — d.h. ein Systemverhalten, das die Existenz mehrerer Replikate eines Objekts verdeckt — zu ermöglichen.

3.1 Eine Strukturierung der für Objekte sichtbaren Zeit

Synchronisation von Operationen ist in einem verteilten System nur beschränkt möglich⁵. Damit dennoch mehrere Benutzer den Eindruck haben, an gemeinsamen Datenobjekten zu arbeiten, müssen künstliche Maßnahmen zur Synchronisation getroffen werden. Das sind z.B. Sperren bei pessimistischen Replikationsverfahren oder Verzögerungen bei gruppenkommunikationsbasierten Verfahren. All dies kann zu Anomalien führen, die Benutzern sichtbar sind, die über weitere, systemexterne Kanäle — z.B. das Telefon — miteinander kommunizieren. Absolute Synchronität in Bezug zur Realzeit kann somit nie Ziel einer generischen Systemunterstützung sein.

Korrektheit kann dann aber auch dadurch erreicht werden, daß das System sicherstellt, daß Operationen zu irgendeinem Zeitpunkt des Systemlaufs hätten durchgeführt werden können. Verschieben von Operationen in die Systemvergangenheit oder die Systemzukunft ist möglich, wenn andere Mitbenutzer diese partielle Zeitverschiebung nicht bemerken können. Auch ein Sprung von der Vergangenheit in die Zukunft ist unter gewissen Randbedingungen korrekt. Einzelheiten sind in [Kot93, Kot95a] zu finden. Wesentliche Randbedingung ist, daß Operationen, die aus Sicht des Benutzers zusammengehören nie zu verschiedenen Zeiten durchgeführt werden.

Die Idee für verschiedene Benutzer verschiedene Zeiten zu wählen, kann nun unter Ausnutzung der Semantik von Objekten eine effektive Basis für pessimistische Replikationsalgorithmen, die für den Kontext mobiler Systeme hinreichend sind, bieten. Zentrales Element ist dabei die *invalidates*-Relation zwischen den Operationen eines Objekts. Intuitiv besagt sie, daß eine Operation op_1 eine Operation op_2 invalidiert, wenn sie deren Ergebnis ungültig machen würde. Eine formale Definition ist in [Kot93] zu finden. Als Beispiel ist in Abbildung 27 die Relation für ein Kundenobjekt, das u. a. Kundennamen, -adressen und -umsatzsalden verwaltet, aufgeführt.

Entsprechend dieser Relation ist es dem System möglich, die Operationen auf dem Objekt in drei Mengen zu gruppieren. Einer Menge **M1**, die nur Operationen enthält, die keine andere invalidieren können, eine Menge **M2**, die nur Operationen enthält, die nicht invalidiert werden können und eine Menge **MA**, die alle Operationen enthält. Für das Beispiel ist diese Aufteilung in Abbildung 28 gezeigt.

⁵ Die natürliche Grenze ist durch die Endlichkeit der Lichtgeschwindigkeit und somit auch des Zusammenwirkens mehrerer Rechnern nach Einsteins spezielle Relativitätstheorie gesetzt.

invalidates

```

setze_plz i1 →() inv lese_plz () →i2
  when i1 != i2;
erhoehe_umsatz i1 →() inv lese_umsatz () →i2
  when i1 != 0;

```

Abbildung 27. Invalidates-Relation

```

M1 = {lese_plz, lese_umsatz}
M2 = {setze_plz, erhoehe_umsatz}
MA = {lese_plz, lese_umsatz
      setze_plz, erhoehe_umsatz}

```

Abbildung 28. Resultierende Mengen

Ein Replikat, das die Operationen der Menge **MA** durchführen darf, existiert in der Systemgegenwart. Da alle Operationen aus **M1** keine sichtbaren Effekte auf andere Benutzer haben, können sie in der Vergangenheit durchgeführt werden. Die Operationen aus **M2** sind hingegen durch keine anderen beeinflussbar, so daß sie immer in die Zukunft delegiert werden können. Eine noch weitere Verfeinerung der Zeitstrukturierung in zyklenfreie Operationsmengen ist möglich.

Um nun Korrektheit zu garantieren, darf im System genau ein Replikat alle Operationen der Menge **MA** durchführen. Dieses Replikat bildet folglich die Systemgegenwart. Operationen aus den Mengen **M1** oder **M2** dürfen von beliebig vielen Replikaten durchgeführt werden, ohne die Korrektheit zu verletzen. Diese Replikate arbeiten also in der sicheren Vergangenheit oder Zukunft des Systems. Das Replikat, das alle Operationen ausführen darf, wird als *Master-Replikat* bezeichnet. Andere Replikate spalten sich von diesem ab, führen abgekoppelt Operationen gemäß **M1** oder **M2** durch, und vereinigen sich später wieder mit dem Rest des Systems.

3.2 Notwendige Synchronisationen

Hat ein Replikat sich auf eine der Mengen **M1** oder **M2** — und somit auf ein Arbeiten in der Vergangenheit oder Zukunft — festgelegt und erhält dann einen Operationsaufruf, der nicht mehr durch diese Menge abzudecken ist, so könnte bei Ausführung der Operation Korrektheit nicht mehr garantiert werden. Nur in dem Fall, daß die erste Operation zu **M1** und die zweite zu **M2** gehört und zudem beide aus Sicht des Benutzers keine zusammengehörige Einheit bilden, kann durch einen Zeitsprung von der Vergangenheit in die Zukunft Korrektheit ohne Kommunikation gewährleistet werden. Ein Sprung von **M2** zu **M1** würde hingegen wieder zu Anomalien führen.

Kann eine Aufruffolge nicht mehr innerhalb der Mengen **M1** oder **M2** befriedigt werden, so stehen dem Replikat die folgenden Wahlmöglichkeiten offen. Neben der Option, die Operation zurückzuweisen (*revoke*), die Operation auf einen späteren Reintegrations-Zeitpunkt zu verschieben (*postpone*), beim Master-Replikat die Gültigkeit der Operation zu sichern (*inquire*) — ist das Master-Replikat nicht erreichbar, so muß auf *revoke* oder *postpone* zurückgegriffen werden — kann die Durchführung der Operation optimistischer Weise riskiert werden, allerdings mit dem Risiko, bei einer späteren Reintegration zu scheitern und manuell nachbessern zu müssen (*risk*). Weiterhin haben Replikate den Freiheitsgrad, ob sie die Menge **M1** (*cheap*) oder **M2** (*queue*) wählen sollen. Zudem ist es möglich, sich nicht a priori festzulegen (*choose*), sondern ein Replikat zu erzeugen, das erst nach Erhalt der ersten Operation die entsprechende Menge „lazy“ wählt.

Das Festlegen der jeweiligen Rechtestruktur ist für jedes Replikat lokal — d.h. unabhängig von den anderen Replikaten — möglich. Allerdings wäre der Zwang, für jedes Replikat Rechte getrennt festlegen zu müssen, für die Praxis zu aufwendig. Deshalb wird ein dreistufiger Ansatz gewählt. Als Default-Rechte werden **rights** = *inquire*(*revoke*) und **set** = *choose* verwendet. Diese Wahl kann durch eine Zuweisung für alle lokalen Replikate einer Klasse überschrieben werden. In der feinsten Granularitätsstufe ist schließlich das Ändern der Rechte individueller Replikate möglich.

3.3 Verwaltung der Replikatpopulation

Anzahl und Zustand der einzelnen Replikate werden durch die folgenden Managementoperationen verwaltet:

Create: Dient zur Erzeugung eines Replikats. Vorlage kann sowohl das Master-Replikat als auch ein beliebiges anderes Replikat sein. Die Durchführung erfordert die Übertragung des Objektzustandes.

Reintegrate: Gleicht den Zustand zweier Replikate ab. Dies ist immer möglich, falls eines der beiden Replikate das Master-Replikat ist. Sonst ist es lokal nur dann möglich, wenn beide dieselbe Operationsmenge gewählt hatten. Bei Beschränkung auf pessimistische Konfliktbewältigung müssen zudem nur die modifizierenden Operationen übertragen werden.

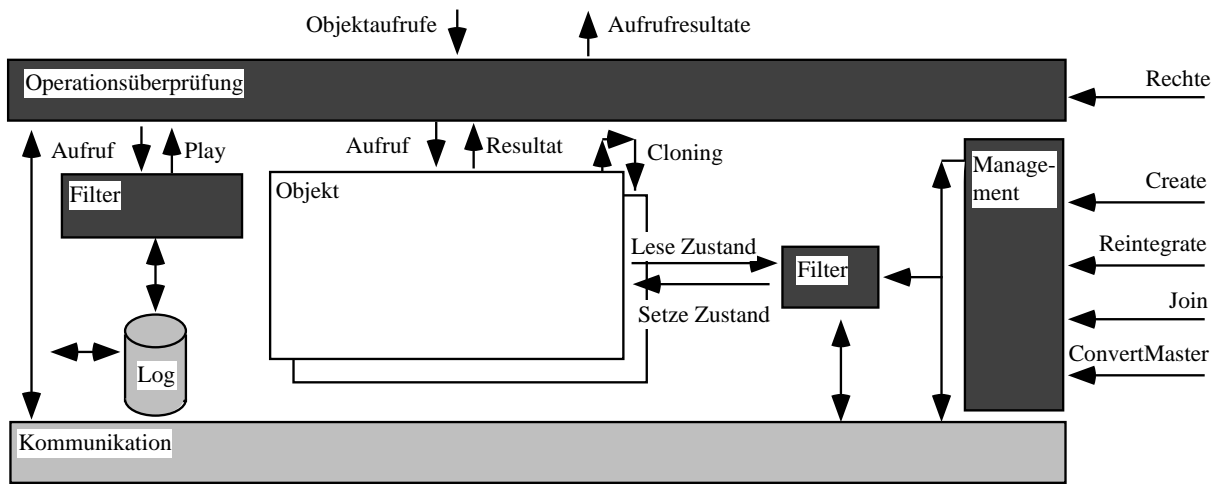


Abbildung 29. Systemarchitektur

Join: Vereinigt zwei Replikate. Entspricht einem Reintegrate unter Aufgabe eines der Replikate. Ist dabei das Master-Replikat nicht erreichbar und arbeiten beide Replikate mit unterschiedlichen Operationsmengen, so wird die Operation auf eine verzögerte Übertragung des Zustands an das Master-Replikat abgebildet.

ConvertMaster: Entspricht einem Reintegrate, wobei eines der beiden Replikate das Master-Replikat ist. Zusätzlich wird der Status Master-Replikat zu sein übertragen.

Die Managementoperationen sind so gestaltet, daß sie immer nur der Kommunikation zwischen zwei (beliebigen) Replikaten bedürfen. Eine schärfere Voraussetzung wäre im Mobile Computing auch sinnvoll. Mit ihnen ist die Nachbildung jeder Managementpolitik möglich.

3.4 Modellierung eines Beispielszenarios

Eine beispielhafte Verwendung der dargestellten Prinzipien erfolge im Kontext des bereits bekannten Kundenobjekts. Eine Firma beschäftige Vertreter, die Kundenbesuche vornehmen. Da der besuchte Kundenkreis bereits am Vortag bekannt ist, lädt jeder Vertreter im Hotelzimmer über eine Modem- oder ISDN-Verbindung die Kundendaten für den kommenden Tag, um über aktuelle Informationen zu verfügen und gegebenenfalls vor Ort notwendige Korrekturen am Datenbestand direkt vornehmen zu können. Unabhängig davon können in der Zentrale Aufträge des Kunden bearbeitet werden, deren Abwicklung den Umsatzsaldo des Kundenobjekts erhöht.

Das Beispielszenario kann mit der vorgestellten Methode wie folgt modelliert werden. Im Hotel wird mittels *Create* ein Replikat der Daten für die am kommenden Tag zu besuchenden Kunden auf dem Mobilrechner des Vertreters erzeugt. Dieses Replikat wird dann mittels *ConvertMaster* zum Master-Replikat. Der Vertreter kann nun beliebige Operationen lokal durchführen. Gleichzeitig ist es der Zentrale möglich, eingehende Aufträge zu buchen, da die *erhoehe_umsatz()*-Operation in der Menge **M2** enthalten ist. All dies ist konfliktfrei allein mit der vorgestellten Default-Rechtestruktur (**rights** = inquire(revoke); **set** = choose) möglich.

4 Prototypische Realisierung

Das dargestellte generische Replikationskontrollverfahren für abgekoppelte Objekte wird aktuell mit zusätzlichen Erweiterungen für optimistische Operationsdurchführung und -konfliktauflösung implementiert. Die dabei gewählte Systemarchitektur ist in Abbildung 29 dargestellt. Alle hellgrauen Komponenten sind generisch realisiert. Die dunkelgrauen Komponenten bauen auf Definitionsdateien auf, in denen die invalidates-, die bei Optimismus benötigte causes-Relation [Kot93] und die Syntaxdefinitionen der Aufruf- und Resultatparameter der Objekte, sowie der Objektstruktur enthalten sind.

Die Architektur ist im wesentlichen durch die unvermeidliche Heterogenität im Mobile Computing geprägt. So baut die Realisierung auf einer einfachen Kommunikationsschnittstelle zwischen benachbarten Replikaten und einer Konvertierung der Daten über einen Filter in ein plattformunabhängiges Format auf. Ein Kommunikationskanal kann selbst unter Einbezug von PDAs immer hergestellt werden.

Einsatz des OSI-Management-Framework im Bereich Anwendungsmanagement

Günter Schäfer

1 Einleitung

Mit der zunehmenden Verbreitung von Rechnernetzen und der damit einhergehenden Entwicklung, die einzelnen Netze auch miteinander zu verbinden, hat sich ein starkes Interesse an technischer Unterstützung für die Verwaltung und den Betrieb komplexer Netzwerke entwickelt. Diesem Interesse wurde in den vergangenen zehn Jahren durch die Entwicklung zahlreicher Konzepte und Architekturen [PT87], [CFSD90], [ISO89] Rechnung getragen. Die *Internationale Organisation für Standardisierung* (ISO) hat im Jahr 1989 als Ergänzung zu ihrem *OSI-Referenzmodell* einen Rahmen für das Management offener Systeme [ISO89] vorgeschlagen. Die darin entwickelten Konzepte gelten weltweit als der langfristig zu favorisierende Ansatz für das Netzwerkmanagement. Mittlerweile ist die Standardisierung der einzelnen Protokolle und Erweiterungen weitgehend abgeschlossen, so daß in den kommenden Jahren mit einem zunehmenden Einsatz der OSI-Konzepte in der Praxis des Netzwerkmanagements zu rechnen ist. Das Netzwerkmanagement kann auf dieser Grundlage als relativ gut entwickelt eingestuft werden.

Unabhängig von dieser Entwicklung sind große Fortschritte im Bereich der Programmierung verteilter Anwendungen gemacht worden, so daß auch hier bereits zahlreiche Konzepte zur Verfügung stehen, welche allmählich im Rahmen der Bemühungen der ISO um *Open Distributed Processing* (ODP) Eingang in den Standardisierungsprozeß finden. Hierbei wird aber anscheinend der Fehler wiederholt, Managementaspekte nicht bzw. nur unzureichend zu betrachten (siehe hierzu [HA93] S. 423 – 426, [ISO93a], [ISO94a], [ISO94b] und [ISO93b]).

Unter dem Begriff Anwendungsmanagement werden im Moment eine Reihe von Forschungsaktivitäten betrieben, welche sich mit unterschiedlichen Teilproblemen wie z.B. Quality of Service [GSFY94], Konfigurationsverwaltung [Zei94] oder Integration von Trading und Managementfunktionalität [KW94] in verteilten Anwendungen beschäftigen. Auf diesen Teilgebieten sind deutliche Fortschritte erkennbar; dennoch ist all diesen Aktivitäten gemein, daß sie das Anwendungsmanagement aus dem jeweiligen Blickwinkel isoliert betrachten und noch nicht in den Gesamtkontext eines integrierten Netzwerk- und Systemmanagements einbetten.

In diesem Aufsatz wird versucht, einige Notwendigkeiten und Implikationen einer solchen Einbettung aufzuzeigen. Hierzu werden zunächst die Grundprinzipien des OSI-Management-Framework dargelegt. Im Anschluß daran wird der Frage nachgegangen, in welchem Maße sich die Konzepte des Netzwerkmanagements auf das Anwendungsmanagement übertragen lassen. Von einer einheitlichen Organisation des Netzwerk- und Anwendungsmanagements verspricht man sich eine Reihe von Vorteilen. Diese auszuloten ist der Gegenstand eines weiteren Abschnittes. Eine zusammenfassende Bewertung und ein Ausblick auf zukünftige Arbeiten schließen den Aufsatz.

2 Grundprinzipien des OSI-Netzwerkmanagements

Im Addendum 4 (Management Framework) zum ISO-OSI-Basisreferenzmodell-Dokument [ISO89] wird OSI Management wie folgt definiert: "The facilities to control, coordinate and monitor the resources which allow communication to take place in the OSI environment". Im OSI-Management-Framework (OSI-NMF) werden die grundlegenden Begriffe und Konzepte des OSI-Netzwerkmanagements (OSI-NM) eingeführt, welche in weiteren Dokumenten verfeinert werden. OSI-NM ist kommunikationsorientiert, d.h. es behandelt im wesentlichen nur die Aspekte, welche sich mit Management durch Kooperation offener Systeme beschäftigen und betrachtet das autonome Management einzelner Systeme als "local matter". OSI-NM wird in 4 Teilmodellen strukturiert:

1. Das *Informationsmodell* liefert einen standardisierten Beschreibungsrahmen für Managementinformation. Die ISO hat hier einen objektorientierten Ansatz gewählt, welcher Datenabstraktion und Vererbungsmechanismen vorsieht. Reale Ressourcen werden durch sogenannte *Managed Objects* (MO's) für das Netzwerkmanagement repräsentiert, die in einer *Management Information Base* (MIB) gespeichert werden.
2. Das *Organisationsmodell* definiert die "Akteure" des OSI-NM und legt ihre Rollen fest. Die ISO sieht hierfür zwei Rollen vor:

- (a) Der *Manager* ist Teil einer Managementanwendung und bietet dieser eine Schnittstelle für das Manipulieren von Managementinformation im Netzwerk.
 - (b) Der *Agent* ist der Kommunikationspartner des Managers und für die Manipulation der MO's zuständig.
3. Das *Kommunikationsmodell* unterscheidet drei verschiedene Managementkategorien:
- (a) *Schichtübergreifendes Management* stellt den allgemeinen Fall des Managements offener Systeme dar und wird durch kooperierende Anwendungsprozesse realisiert. Die Kooperation der Prozesse erfordert Kommunikation zwischen ihnen. Diese Kommunikation wird über sogenannte *System Management Application Entities* (SMAE) abgewickelt, welche sich auf eigens dafür entwickelte *Common Management Information Services* (CMIS) abstützen können. Zur Erbringung der CMIS-Dienste wird das *Common Management Information Protocol* (CMIP) verwendet.
 - (b) *Schichtenmanagement* beschränkt sich auf schichtenspezifische Managementaspekte, für die keine Dienste höherer Schichten benötigt werden. Beispiele hierfür sind Schleifentests oder der Austausch von Routinginformation.
 - (c) *Protokollmanagement* behandelt protokollinherente Managementaspekte, wie z.B. Fenstergrößen, Timer oder Protokollparameter der Verbindungsaufbauphase.
4. Das *Funktionsmodell* des OSI-NM unterscheidet die folgenden fünf Funktionsbereiche:
- (a) Das *Konfigurationsmanagement* unterstützt den Netzwerkverwalter bei der Einstellung von Parametern eines offenen Systems. Hierzu ermöglicht es ihm, Managed Objects zu erzeugen, zu manipulieren und zu löschen.
 - (b) Das *Fehlermanagement* stellt Funktionen für die Fehlererkennung und -behebung zur Verfügung.
 - (c) Im *Leistungsmanagement* werden statistische Daten gesammelt, aus denen Aussagen über das Leistungsverhalten eines Netzes hergeleitet werden können, um es zu optimieren.
 - (d) Das *Abrechnungsmanagement* ist für die Protokollierung und Abrechnung der Ressourcennutzung im Netzwerk zuständig.
 - (e) Der Bereich *Sicherheitsmanagement* stellt Mechanismen für die Authentisierung, Authorisierung und die Schlüsselverwaltung bereit.

Zur Unterstützung der Funktionsbereiche sind von der ISO eine Reihe allgemeiner Management-Hilfsfunktionen, die sogenannten *System Management Functions* (SMF), entwickelt worden. Die SMF's sind meist generisch definiert und können von den Funktionsbereichen nach den jeweiligen Anforderungen parametrisiert und genutzt werden.

Mit dem OSI-NMF ist eine sehr mächtige Architektur definiert worden, die hier nur ansatzweise skizziert werden kann. Besonderes Augenmerk soll an dieser Stelle auf das Informations- und das Organisationsmodell gerichtet werden: der von der ISO gewählte objektorientierte Ansatz bietet mit seinen Möglichkeiten der Klassenbildung, Mehrfachvererbung, Allomorphie und Enthaltenseinsrelation umfangreiche Mittel zur Strukturierung von Managementinformation. Weiterhin erlauben die Common Management Information Services es, Operationen auf Teilmengen von Managed Objects, welche durch Bereichseingrenzung und Filterung selektiert werden können, synchronisiert durchzuführen.

3 Einsatz des OSI-NMF im Bereich Anwendungsmanagement

Die in Abschnitt 2 zitierte Definition des OSI-NM beschränkt den intendierten Wirkungsbereich des OSI-NMF deutlich auf kommunikationsorientierte Aspekte in offenen Systemen. Diese Entscheidung ergibt sich aus der Zielrichtung der OSI-Standardisierung, Verbindungen zwischen offenen Systemen zu ermöglichen. Ein Einsatz der hier erarbeiteten Konzepte im Bereich Anwendungsmanagement ist eigentlich nicht das Ziel der Arbeitsgruppen gewesen, da bei dem Management verteilter Anwendungen nicht Verbindungen zwischen offenen Systemen sondern Aspekte der Verarbeitung über Rechnergrenzen hinweg im Zentrum des Interesses stehen. Verarbeitung über Rechnergrenzen hinweg ist zentrale Thema der Standardisierungsbemühungen um Open Distributed Processing.

Dennoch gilt es zu bedenken, daß die Ergebnisse der Standardisierung des OSI-NM eine enorme wirtschaftliche Investition für die gesamte informationsverarbeitende Industrie darstellen. Bereits aus diesem Grund empfiehlt es sich, die Möglichkeiten einer Integration von Anwendungs- und Netzwerkmanagement sorgfältig zu untersuchen. Wie Abschnitt 4 zeigen wird, verspricht eine integrierte Behandlung dieser beiden Bereiche noch weitere Vorteile.

Zunächst einmal soll jedoch die Eignung des OSI-NMF für das Anwendungsmanagement im Mittelpunkt des Interesses stehen. In den folgenden Abschnitten werden deshalb die einzelnen Modelle des OSI-NMF unter dieser Fragestellung näher betrachtet.

3.1 Das Informationsmodell

In [Ash93] ist das MO-Objektmodell der ISO mit dem Objektmodell der *Object Management Group* (OMG) verglichen worden. Ziel dieses Vergleiches war es, die Gemeinsamkeiten und Unterschiede der beiden Modelle im Hinblick auf die Implementierung von Managed Objects mit den Methoden und Werkzeugen der OMG auszuloten. Hierbei wurden fundamentale Gemeinsamkeiten aber auch einige Unterschiede und Gegensätze festgestellt. Im gleichen Bericht werden bereits Maßnahmen vorgestellt, wie die Gegensätze und Unterschiede harmonisiert und miteinander vereinbart werden können. In der abschließenden Zusammenfassung der Kommission werden die Chancen der Harmonisierung sehr positiv bewertet und den objektorientierten Entwicklungsmethoden eine signifikante Rolle bei der Entwicklung von Managementprodukten auf der Basis von OSI-NM-Standards zugesprochen.

Die weitgehende Übereinstimmung der beiden Objektmodelle spricht aber auch für einen Einsatz des OSI-NM-Informationsmodells in dem Bereich Anwendungsmanagement. Sie erzeugt zumindest begründete Hoffnung dafür, daß sich für objektorientiert entwickelte Software relativ leicht Managed Objects definieren und realisieren lassen. Das OSI-NM-Informationsmodell erscheint aus diesem Grund als geeignet für das Anwendungsmanagement.

3.2 Das Organisationsmodell

Das Organisationsmodell des OSI-NM sieht die beiden Rollen des Managers und des Agenten vor. Diese Rollen sind nicht statisch festgeschrieben. Es ist daher möglich, daß eine Instanz sowohl in der Rolle des Managers als auch in der Rolle des Agenten auftritt. Diesbezüglich ist das Modell also hinreichend flexibel.

Eine interessante Frage in diesem Zusammenhang ist, ob und wie sich Agentenfunktionalität in verteilte Anwendungen integrieren läßt. Hierzu ist zu bemerken, daß Managementimpulse sowohl von einem Manager als auch von Managed Objects ausgelöst werden können. Dem Agenten ist daher in gewissem Sinne eine vermittelnde Funktion beizumessen. Es empfiehlt sich daher die folgende Realisierungsvariante: der Agent wird als eigenständige Instanz ausgelegt, welche über (mindestens) zwei Kanäle Impulse empfängt. Ein Kanal dient der Kommunikation in Richtung des Managers, und ein weiterer Kanal ist für Meldungen von und an Managed Objects zuständig. Der Agent sollte sinnvollerweise auf dem gleichen Rechner wie die verwaltete Anwendung (*Managed Application*, MA) laufen, damit er Managed Objects lokal manipulieren kann. Bei verteilten Anwendungen muß der Agent daher repliziert werden. Eine MA wird also in der Regel von einer Menge von Agenten managementseitig unterstützt. Die Agenten dieser Menge kooperieren miteinander, um Anfragen und Operationen von Managern zu bearbeiten und Meldungen an diese weiterzuleiten. Diese Kooperation kann teilweise⁶ mit den Methoden des OSI-NMF realisiert werden, da die Manager- und Agentenrollen ja nicht statisch vorgegeben sind und Agenten somit fallweise in eine Managerrolle gegenüber anderen Agenten schlüpfen können.

3.3 Das Kommunikationsmodell

Von den drei Managementkategorien des OSI-NM-Kommunikationsmodells ist lediglich das schichtübergreifende Management für den Bereich Anwendungsmanagement von Interesse. Die hier zur Verfügung stehenden Common Management Information Services stellen mit ihren Möglichkeiten zur Selektion von Managed Objects ein sehr mächtiges Hilfsmittel dar. So können ganze Unterbäume oder auch einzelne Ebenen von Unterbäumen des *Management Information Trees* (MIT) für eine Operation ausgewählt werden, aus denen einige Managed Objects durch Angabe von Attributseigenschaften herausgefiltert werden können.

Desweiteren kann für eine Operation festgelegt werden, ob diese atomar, d.h. nur von allen selektierten MO's oder gar nicht, oder bestmöglich, also von so vielen selektierten Managed Objects wie möglich, ausgeführt werden soll. Hier ergibt sich ein Realisierungsproblem: eine verteilte Anwendung wird, wie bereits in Abschnitt 3.2 ausgeführt, von einer Menge von Agenten kooperativ verwaltet. Wie soll nun ein Agent agieren, wenn er den Auftrag erhält, eine Operation atomar auf einer Menge von MO's auszuführen, diese aber nicht alle von ihm lokal verwaltet werden? Muß lediglich ein weiterer Agent von ihm konsultiert werden, so kann er zunächst diesen die Operation atomar ausführen lassen und im Erfolgsfall die Operation lokal ausführen. Wird die Unterstützung mehrerer Agenten benötigt, so erweisen sich die Common Management Information Services als unzureichend. Eine mögliche Lösung ist in diesem Fall der Einsatz der *Commitment, Concurrency and Recovery* (CCR) oder der *Transaction Processing* (TP)

⁶ Siehe hierzu auch den folgenden Abschnitt 3.3

Dienstelemente der Anwendungsschicht. Dies impliziert, daß die Kooperation von Agenten nicht immer ausschließlich mit den Methoden des OSI-NMF bewerkstelligt werden kann.

Man sollte das aber nicht überbewerten und dem OSI-NMF allein deshalb die Eignung für das Anwendungsmanagement absprechen. Ein wesentlicher Punkt des Anwendungsmanagements ist es ja, eine Management-Schnittstelle für a priori, d.h. bei der Erstellung der Managed Application, unbekannte Managementanwendungen zur Verfügung zu stellen. Dieser Forderung können die Common Management Information Services aber durchaus gerecht werden.

3.4 Das Funktionsmodell

Das Funktionsmodell des OSI-NMF unterscheidet die in Abschnitt 2 aufgezählten Funktionsbereiche. Diese sind so allgemein definiert, daß sich wohl die meisten Funktionen des Anwendungsmanagements zu einem der Bereiche zuordnen lassen werden. Viel interessanter ist in diesem Zusammenhang die Frage, welcher Nutzen aus den bereits für das Netzwerkmanagement definierten System Management Functions und den dazugehörigen *Support Managed Object Classes* für das Anwendungsmanagement gezogen werden kann und für welche Aufgaben eventuell neue SMF's geschaffen werden müssen. Hierbei zeigt es sich, daß einige der bereits definierten SMF's, wie z.B. die *Alarm Reporting Function*, die *Event Report Management Function*, die *Security Audit Trail Function*, die *Objects and Attributes for Access Control* oder die *Workload Monitoring Function* auch im Anwendungsmanagement von Nutzen sein können und somit Entwicklungsaufwand eingespart werden kann, wenn auf sie zurückgegriffen wird.

4 Vorteile eines integrierten Netzwerk- und Anwendungsmanagements

In den vorangehenden Abschnitten stand die Eignung des OSI-NMF für das Anwendungsmanagement im Mittelpunkt der Betrachtung. Über die Eignung hinaus sind aber vor allem die Vorteile einer integrierten Behandlung von Netzwerk- und Anwendungsmanagement von Interesse.

Als Hauptvorteil ist hier zu nennen, daß auf der Managerseite wahrscheinlich keine neuen Konzepte benötigt werden und somit auch bereits entwickelte Managementplattformen und -oberflächen genutzt werden können. Dies bedeutet zum einen weniger Entwicklungs- aber auch weniger Einarbeitungsaufwand bei einer Einführung des Anwendungsmanagements in eine bestehende Netzwerkinstallation.

Desweiteren ermöglicht ein integrierter Ansatz Managementanwendungen, die mit isolierten Techniken nicht oder nur umständlich erstellt werden können, wie z.B. eine Lastverteilung mit integrierten Accounting-Mechanismen, die den Benutzer einer verteilten Anwendung an den Kosten, welche die Lastverteilungsmaßnahmen erzeugen, beteiligt. Denkbar sind in diesem Zusammenhang beliebige Managementanwendungen, welche Aspekte des Netzwerkmanagements mit solchen des Anwendungsmanagements verbinden. An dieser Stelle sollte auch bedacht werden, daß eine vom Netzwerkmanagement losgelöste Betrachtung des Anwendungsmanagements wahrscheinlich nur zu unvollständigen Ergebnissen kommen wird, da wichtige Größen wie z.B. die Performance oder die Ausfallsicherheit einer verteilten Anwendung ja im besonderen Maße auch vom aktuellen Zustand des zugrundeliegenden Netzwerkes abhängen.

5 Zusammenfassung und Ausblick

In diesem Aufsatz sind Grundfragen der Eignung des OSI-Network-Management-Frameworks für das Anwendungsmanagement erörtert worden. Eine Betrachtung der einzelnen Teilmodelle des OSI-NMF hat ergeben, daß es durchaus vielversprechend erscheint, Anwendungsmanagement nach den Prinzipien und mit den Methoden des Netzwerkmanagements zu organisieren. Die Hauptvorteile eines solchen homogenen Ansatzes werden in kürzeren Entwicklungszeiten und in einem geringeren Einarbeitungsaufwand für Managementanwendungen gesehen.

In zukünftigen Arbeiten werden die in diesem Aufsatz aufgestellten Thesen kritisch zu prüfen sein. Hierzu soll zunächst ein Beispielszenario entworfen werden, in welchem eine konkrete verteilte Anwendung mit Agentenfunktionalität ausgestattet und von einer Managementanwendung verwaltet wird. Dabei können bereits unterschiedliche Einbettungen der Agenten in die verteilte Anwendung (d.h. Realisierung als eigener Prozeß, Thread oder Unterprogramm) erprobt werden. Sollte sich der Ansatz als praktikabel erweisen, so werden in weiteren Arbeiten allgemein verwendbare Managed Objects und eventuell Vorschläge für System Management Functions für das Anwendungsmanagement erarbeitet. Bei der Entwicklung einer verteilten Anwendung sollten Managementaspekte bereits möglichst früh berücksichtigt werden. Daher sind Maßnahmen zur Entwurfs- und Entwicklungsunterstützung von verteilten Anwendungen mit Agentenfunktionalität auch ein Ziel unserer Arbeiten auf diesem Gebiet.

CHIMPSY: Ein modulares Prozessorsystem für die Hochleistungskommunikation

Jochen Schiller

Aufkommende Anwendungen benötigen in den meisten Fällen außer einer hohen Leistungsfähigkeit eine große Anzahl unterstützter Kommunikationsdienste. Fortgeschrittene Netzwerke, z.B. ATM-Netze, sind in der Lage, diese Anforderungen durch die Bereitstellung einer sehr hohen Übertragungskapazität und verschiedener Dienste zu unterstützen. Ein Problem liegt nun darin, daß derzeit existierende Kommunikationssubsysteme diese Leistungsfähigkeit nicht an die Anwendungen weitergeben können. Verschiedene Ansätze der letzten Jahre beruhen auf Software-Optimierungen, Parallelverarbeitung oder Hardware-Unterstützung von leistungskritischen Komponenten [ITN92], [SDW92], [Fel94], [SP91], [Bra93], [BZ93]. Allerdings wurde der Einsatz von dedizierter Hardware bisher auf sehr kleine und relativ einfache Komponenten beschränkt [KKS93], [BY92]. Ein wesentlicher Gesichtspunkt bei der Leistungssteigerung spielt die Integration der Kommunikationsaufgabe in das Gesamtsystem [Ste94]. Da die Anwendungsanforderungen nicht statisch sind, sondern sich im Laufe der Zeit weiterentwickeln, ist eine flexible Strategie zum Entwurf komplexer Kommunikationssysteme von großer Bedeutung. Mehrere existierende Ansätze beschäftigen sich mit der formalen Beschreibung eines Kommunikationssystems und der Ableitung einer Implementierung davon [KKS93], [LO94], [KEG93], [PZ92], [SJ93], [WOB93]. Jedoch ist es bis heute nicht gelungen, eine *leistungsfähige* Implementierung für Protokolle höherer Schichten ($\geq 2b$) direkt von einer Spezifikation abzuleiten. Das hier vorgestellte System *CHIMPSY* (Communication oriented High-performance Modular Processor System) adressiert genau die oben angesprochenen Problembereiche: die Implementierung komplexer Protokolle höherer Schichten auf einer leistungsfähigen und flexiblen Architektur, eine hohe Flexibilität bei der Integration von Teilkomponenten in ein Gesamtsystem und die Unterstützung einer formal abgesicherten Abbildung von Kommunikationsprotokollen auf eine Implementierungsarchitektur.

1 Implementierungsarchitektur

Der sog. Verbindungsprozessor (Connection Processor, *CP*) stellt die zentrale Komponente der Architektur dar (Abb. 30, [BSZ94a], [BSZ94b]). Er bildet die Rahmenimplementierungsarchitektur für die Komponenten C_i und die zentrale Verbindungskomponente *CE*. Jeglicher Datenaustausch innerhalb des *CP* und mit externen Komponenten läuft ausschließlich über *CE*. Um eine größtmögliche Flexibilität sicherzustellen, besitzen die Komponenten C_i identische Schnittstellen, ebenso ist das Protokoll zum Datenaustausch zwischen Komponenten stets dasselbe. Das Verbindungselement *CE* stellt zwei unidirektionale Verbindungen jeder Komponenten zur Verfügung, zusätzlich besteht die Möglichkeit der Gruppenkommunikation. Die Komponente *CE* stellt sicher, daß sich Verbindungen zwischen verschiedenen Komponenten nicht gegenseitig beeinflussen. Aus diesen Gründen kann die Implementierung als einfacher Kreuzschienenverteiler ähnlich zu [Tex94] oder als Hochgeschwindigkeitsbus mit Zeitmultiplexzugriff implementiert werden.

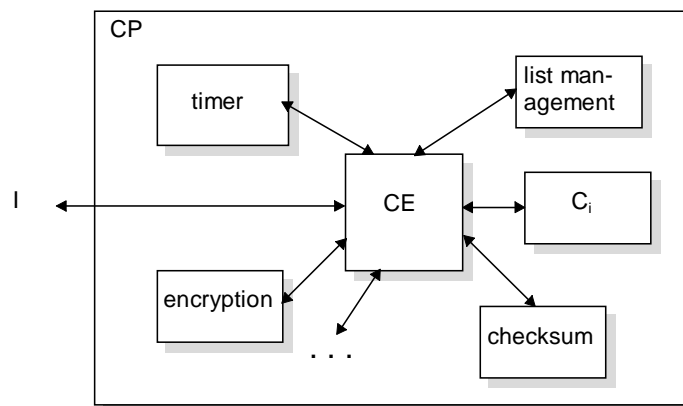


Abbildung 30. Modulare Implementierungsarchitektur des Kommunikationsprozessors

- **Erweiterte endliche Automaten** Zur Unterstützung zeitkritischer Protokollautomaten können spezielle erweiterte endliche Automaten (finite state machine, *FSM*) in einen *CP* integriert werden [SB93], [Sch94]. Auf diese können nun die Automaten einer formalen Protokollspezifikation abgebildet werden. Prinzipiell könnten beliebige Automaten abgebildet werden, jedoch kann eine hohe Effizienz der Implementierung nur bei Protokollen gewährleistet werden, die schon von Anfang an als Menge parallel arbeitender erweiterter endlicher Automaten spezifiziert wurden (z.B. Patrolos [Bra93], [BZ93], XTP [SDW92] oder SNR [SNR90]). Die FSM-Einheiten arbeiten unabhängig voneinander, die Kommunikation geschieht mittels asynchroner Signalen. Auf diese Weise unterstützt die Implementierungsarchitektur die vom Protokollentwerfer vorgesehene Parallelität verschiedener Protokollfunktionen. Jede FSM-Einheit besteht aus einer *Eingangswarteschlange*, der *Steuereinheit* des Automaten und einer lokalen *ALU*. Diese ALU ist den Bedürfnissen des Automaten angepaßt und kann von einem einfachen Addierer bis hin zu einer komplexen Einheit zur Verwaltung dynamischer Datenstrukturen reichen. Die reihfolgegetreue Warteschlange dient der Entkopplung der einzelnen Komponenten voneinander, alle Signale an den Automaten werden in diese eingereiht. Damit entspricht dieser Mechanismus dem der Prozesse der Spezifikations- und Beschreibungssprache SDL der ITU-T [Hog89], [CCI89]. Diese Komponenten stellen also eine Hardware-Implementierungsarchitektur für Protokolle dar, die als eine Menge von SDL-Prozessen spezifiziert wurden.
- **Zeitgeber** Viele Protokollfunktionen, wie z.B. Übertragungswiederholung oder Verbindungsverwaltung, benötigen Zeitgeber. In herkömmlichen Implementierungen werden diese meist vom Betriebssystem zur Verfügung gestellt, was oft einen großen Leistungengpaß darstellt [VL87]. Die hier entwickelte Komponente verwaltet eine dynamische Liste mit Zeiteinträgen und stellt einen Satz von Operationen zu deren Manipulation zur Verfügung. Darunter sind Befehle zum Setzen, Lesen, Löschen und Rücksetzen von Zeitgebern. Nach Ablauf der eingestellten Zeitspanne sendet die Einheit selbständig ein Alarmsignal an die betreffende(n) Komponente(n). Die zeitlich Auflösung beträgt 100ns, die maximale Zeitspanne 7 Minuten, größere Spannen sind im allgemeinen nicht zeitkritisch und sollten in Software implementiert werden.
- **Listenverwaltung** Aufgrund der Komplexität der Manipulationsfunktionen wurde die Datenstruktur der Liste seither kaum in Hardware-Implementierungen benutzt. Jedoch benötigen viele Protokollfunktionen Listen, um dynamische Daten zu speichern. Ein zeitkritisches Beispiel ist die Unterstützung der selektiven Übertragungswiederholung, hierfür wurde eine spezialisierte Komponente entwickelt [CS95], [SZ95a]. Angenommen eine Anwendung verlangt die Überwachung jedes abgesendeten Pakets, die Übertragungskapazität beträgt 1Gbps, die Paketlänge 1000Bytes, so muß etwa alle $8\mu s$ eine Änderung an der Liste vorgenommen werden. Dies ist mit reinen Software-Implementierungen praktisch unmöglich. Die hier entwickelte Komponente unterstützt zusätzlich Gruppenkommunikation, was eine neue Dimension der Komplexität mit sich bringt [CSS94]. Zur Manipulation der Liste stehen unter anderem die in Tabelle 2 dargestellten Befehle zur Verfügung.

Befehl	Eingabeparameter	Ausgabeparameter	Kommentar
init_list	rec_id, seq_no		Initialisierung einer neuen Liste für die Verbindung rec_id mit der initialen Bytesequenznummer seq_no
add_mcg	mc_con_id, rec_id		Hinzufügen des Empfängers rec_id zur Multicast-Gruppe mc_con_id
set_rel	mc_con_id, k		Setzen der Zuverlässigkeitsklasse für die Multicast-Gruppe mc_con_id auf k
set_gap	rec_id, seq_no, length		Eintrag des Pakets (seq_no, seq_no + length); Überlappende Einträge werden automatisch verbunden bzw. gelöscht
del_gap	rec_id, seq_no, length		Löschen eines Eintrags bzw. Teile eines Eintrags der Form (seq_no, seq_no + length)
get_gap	rec_id, ptr	seq_no, length, next	Auslesen der Daten für den Empfänger rec_id auf die ptr zeigt, next zeigt auf den nächsten vorhandenen Eintrag der Liste

Tabelle 2. Beispiele für Befehle der Listenverwaltung

2 Systementwurf

Einer der wichtigsten Aspekte beim Entwurf umfassender Systeme ist die nahtlose Einbettung aller Teilkomponenten. Bis heute jedoch ist diese Integration bezüglich der Kommunikationskomponente nicht zufriedenstellend gelöst worden. Traditionellerweise bestehen Kommunikationskomponenten aus einem Netzwerkadapter, der mit dem Systembus verbunden ist, und mehreren Programmen, die auf dem Standardprozessor abgearbeitet werden. Aus diesem Grund leiden alle diese Ansätze unter einer schlechten Systemleistung, die durch eine hohe Anzahl Datenkopieraktionen, Buszugriffskonflikte, die Speicherhierarchie und Serialisierung der Programmabarbeitung hervorgerufen wurde. So sieht z.B. eine Anwendung bei einer *ansonsten unbelasteten* Workstation⁷ lediglich 55Mbps eines 100Mbps ATM-Adapters (hierbei beträgt die CPU-Auslastung bis zu 80%!). Aus diesen Gründen wurde CHIMPSY von Anfang an so entworfen, daß die Komponenten in unterschiedlicher Weise in ein Gesamtsystem eingebettet werden können.

- **Koprozessoransatz:** Ähnlich der Unterstützung der Ganzzahlarithmetik einer Standard-CPU durch einen Fließkommaarithmetik-Koprozessor, kann ein *CP* die CPU durch die Abarbeitung zeitkritischer Protokollfunktionen unterstützen. Der Vorteil dieses Ansatzes liegt darin, daß nur kleine Änderungen an bestehenden Programmen vorgenommen werden müßten. So kann mit Hilfe eines *CP*-Emulators die gleiche Software auf Rechnern mit oder ohne Hardware-Unterstützung laufen. Anstelle einer Software-Routine wird bei Vorhandensein eines *CP* einfach dessen Mikroprogramm aufgerufen. Die Nachteile der zugrundeliegenden Rechnerarchitektur werden dadurch natürlich nicht beseitigt, jedoch können erfolgreiche Software-Optimierungsmethoden, z.B. ILP (Integrated Layer Processing) und ALF (Application Layer Framing) [CT90a], auch weiterhin gewinnbringend eingesetzt werden.
- **Multiprozessoransatz:** Die meisten der heute verkauften Workstations sind Multiprozessorsysteme. Videoprozessoren und mehrere CPUs sind mit Bussen verbunden und teilen sich gemeinsame Daten, deshalb ist es nur konsequent, diese Idee auch auf den *CP* auszuweiten. Der große Vorteil ist die vergleichsweise einfache Integration und der gleichberechtigte Zugriff auf Systemressourcen, jedoch kann hierdurch je nach Rechnerarchitektur z.B. der Flaschenhals 'Speicher' oder 'Systembus' verstärkt werden.
- **Intelligenter Netzadapter:** Der traditionelle Ansatz zur Steigerung der Kommunikationsleistung ist die Erweiterung der Funktionalität und Erhöhung der Leistungsfähigkeit des Kommunikationsadapters. Die Vor- und Nachteile sind ähnlich der des Multiprozessoransatzes, allerdings ist die Integration des Adapters im allgemeinen lockerer, da er oftmals über den langsameren Systembus mit den restlichen Komponenten verbunden wird und auch nicht gleichwertig vom Betriebssystem berücksichtigt werden kann. Der große Vorteil liegt darin, daß lediglich ein neuer Treiber für den Adapter angefertigt werden muß, an den Anwendungen müssen keine Änderungen vorgenommen werden [DPD94].
- **Autonome Einheiten:** Ein radikaler und vielversprechender Ansatz sieht alle Systemkomponenten als autonome Einheiten, die über eine schnelle Verbindungskomponente miteinander gekoppelt sind (z.B. DAN, Desk Area Network [McA93]). Diese Architektur kann die Parallelität zwischen verschiedenen Datenströmen ausnutzen, so kann z.B. gleichzeitig ein Videodatenstrom vom Netzadapter zum Bildschirm und Plattenspeicher geschickt werden während eine der CPUs Daten vom Hauptspeicher liest. Der Nachteil des Ansatzes liegt natürlich in der damit verbundenen radikalen Änderung der Gesamtsystemarchitektur.

2.1 Entwurfsablauf

Ein zentrales Ziel der Forschung ist die automatische Ableitung leistungsfähiger Systeme aus einer formalen Spezifikation. Bei CHIMPSY besteht eine *Spezifikation* aus dem *Kommunikationsprotokoll* und einer Menge von *Konfigurierungsparametern*. Diese Parameter umfassen die Integrationsalternative, Technologie, Schnittstellen und andere Werte; hiervon wird ein *Implementierungsrahmenwerk* abgeleitet. Das Protokoll besteht aus protokollabhängigen und -unabhängigen *Funktionen*. Erstere umfassen die Zustandsübergänge und zugehörige Aktionen wie sie z.B. als SDL-Spezifikation beschrieben werden. Unabhängige Funktionen sind z.B. Zeitgeber, Speicherverwaltung oder Listenverwaltung. Alle Funktionen werden in Form von elementaren *Bausteinen* implementiert. Diese sind flexibel durch Programmier- und Parametrisierbarkeit. Abhängig von der Spezifikation werden nun diese Bausteine in das Implementierungsrahmenwerk zu einem leistungsfähigen *Kommunikationssystem* integriert.

⁷ HP 735, TCP/IP als Protokoll, TCP Fenstergröße 51kbyte

2.2 Entwurfsbeispiel

Das folgende Beispiel zeigt nur einen kleinen Ausschnitt aus dem gesamten Entwurf und zwar die Abbildung einer Zeile (Tabelle 3) einer Protokollautomatenspezifikation in den entsprechenden Mikrocode (Tabelle 4) für die FSM-Einheit eines *CP*. Beim Empfang des Signals *i.rec_closing* im Zustand *ACTIVE* sendet der Automat in der dargestellten Zeile ohne Überprüfung einer Bedingung das Signal *i.rec_seq* an den Automaten *TI* und verharrt im Zustand *ACTIVE*.

Zustand	Ereignis	Bedingung	Aktion	nächster Zustand
ACTIVE	i.rec_closing		signal(TI, i.rec_seq)	ACTIVE

Tabelle 3. Beispiel für eine Zeile einer Protokollspezifikation

Daten	Auswahl	Befehl	Steuerung	Kommentar
move in, base		CNT	GETH	warte auf Signal
		CNT	MOVE S,A	
		BRV	MOVE H1,A	
1		CNT	MOVE S,C	event:
TI; i.rec_seq		CNT	SGNL	i.rec_closing
OFFSET; move mem, out		CNT	MOVE A,Q	
ACTIVE		CNT	SAVE	
label		JMP	SLEEP	

Tabelle 4. Beispiel für Mikrocode einer FSM-Einheit

3 Ergebnisse

Die gesamte Hardware-Architektur wurde mit Hilfe der Sprache VHDL [IEE87] beschrieben, die Simulation und Synthese mit kommerziellen Werkzeugen durchgeführt [Syn94]. Zur Verifikation der Bausteine wurde ihr Verhalten formal als Menge von Automaten beschrieben und eine Spezifikation in temporaler Logik hinzugefügt. Die Verifikation erfolgte mit Hilfe des Programms SMV (Symbolic Model Verifier [McM93], das auf der Methode der symbolischen Modellprüfung beruht [Lut95].

3.1 FSM-Einheit

Die Implementierung der FSM-Einheit besteht aus einer *Steuer-* und einer *Ausführungskomponente* und einem *Speicher* zum Ablegen der Protokollbeschreibung. Bei der Systeminitialisierung wird das entsprechende Mikroprogramm geladen, danach wartet der Baustein auf das erste Signal. Alle Aktionen, wie Adressberechnungen, arithmetische- und logische Operationen und Datenkopieraktionen werden vollständig autonom vorgenommen. Die Schaltung wurde für die Unterstützung von 1024 simultanen Verbindungen mit Hilfe einer $1.0\mu\text{m}$ CMOS-Bibliothek synthetisiert. Daraus resultierte ein Flächenbedarf von 1.78mm^2 und ein kritischer Pfad für die Steuerlogik von 10ns (d.h. es kann eine maximale Frequenz von 100MHz erreicht werden). Bei der Verifikation wurden auf Sender- und Empfängerseite je 7 Automaten des Protokolls PATROCLOS modelliert und Szenarien wie Verbindungsaufbau und Datenübertragung überprüft. Hier konnten dann vielfältige Lebendigkeits- und Fairness-Eigenschaften der Implementierung bewiesen werden. Ein Beispiel ist der Ausdruck $\text{AG}(\text{wr_req} \rightarrow \text{EF wr.state=connect_conf})$. Er bedeutet, daß nach einer Verbindungsaufbauanforderung des Senders (*wr_req*) immer (*AG*, always global) irgendwann einmal (*EF*, exists finally) eine Verbindungsaufbau-bestätigung des Empfängers (*connect_conf*) eingeht. Bei geschickter Modellierung können Beweise dieser Art innerhalb weniger Sekunden auf Workstations durchgeführt werden.

3.2 Übertragungswiederholungs-ALU

Die spezielle ALU zur Listenverwaltung (Abb. 31), angepaßt auf die Übertragungswiederholung, wurde mit Hilfe der Werkzeuge Synopsys und Cadence auf eine $0.7\mu m$ CMOS-Bibliothek bis auf Layout-Ebene abgebildet. Die Schaltung umfaßt 10945 Standardzellen, benötigt für die Steuerlogik eine Fläche von $49mm^2$ und besitzt einen kritischen Pfad von $45ns$. Auf dieser Schaltung gibt es spezielle Logik zum Durchsuchen von Listen, so kann z.B. der Vergleich $a \leq b \leq c, b \leq c \leq d$ innerhalb eines Taktzyklus abgearbeitet werden (zum Vergleich: der DEC Alpha-Chip benötigt für nur einen Vergleich $72.6ns$ bei $6.6ns$ Zykluszeit). Das bedeutet, daß dieser Baustein aufgrund seiner dedizierten Architektur wesentlich schneller als modernste Prozessoren ist, bei denen solche Operationen ja aus vielen Maschinenbefehlen bestehen.

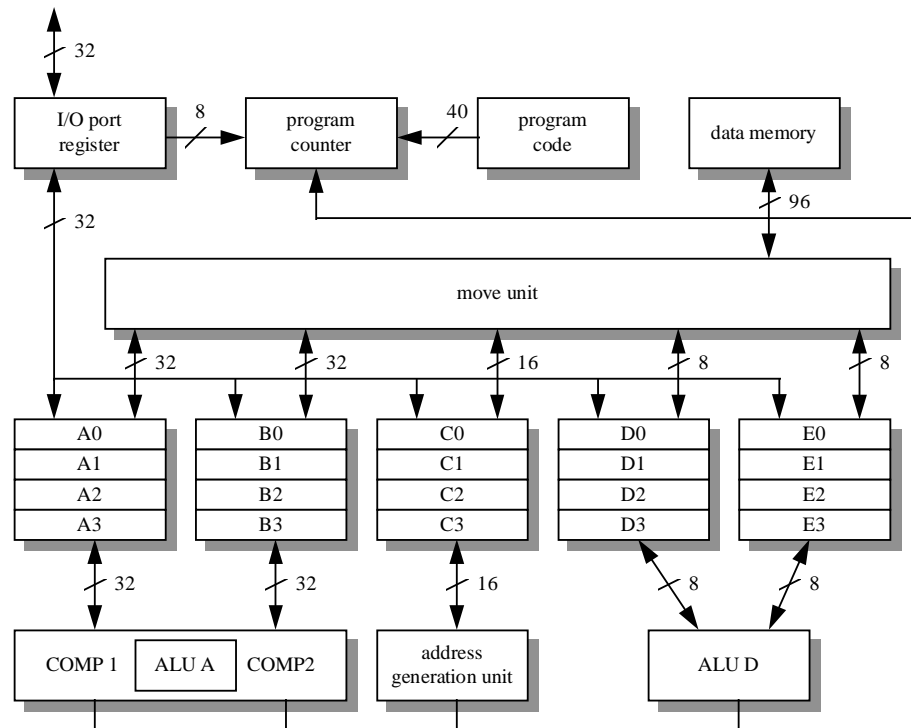


Abbildung 31. Interne Struktur der Übertragungswiederholungs-ALU

4 Ausblick

Weitere Arbeiten umfassen die formale Verifikation zusätzlicher Komponenten, den Entwurf eines SDL-nach-VHDL-Compilers und weitere Synthese und Simulation von Bausteinen. Der SDL/VHDL-Compiler wird speziell auf die entworfene Architektur ausgelegt und somit ein mächtiges Werkzeug bei der Umsetzung von SDL-Spezifikationen in leistungsfähige Implementierungen sein. Parallel zu der Synthese von hochintegrierten CMOS-Schaltungen werden Prototypen der Bausteine mit Hilfe sog. FPGAs (Field Programmable Gate Arrays) erstellt. Dazu ist ein beliebig umprogrammierbares FPGA-Board in einer Sun-Workstation vorhanden, so daß vom Entwurf einer Schaltung bis zu deren Implementierung lediglich wenige Minuten benötigt werden und dies ohne jeglichen Eingriff in die Hardware. Der Baustein kann anschließend bequem über eine C-Schnittstelle angesprochen werden, Programme können Schaltungen auf dem Chip vergleichbar mit Software-Routinen 'aufrufen'. Zusätzlich zu Simulationen der einzelnen Bausteine sind Simulationen kompletter Multimedia-Endsysteme mit Hilfe leistungsfähiger Simulationswerkzeuge im Entstehen. Hier wird dann die entworfene Architektur unter realen Lastbedingungen getestet.

Anwendungsorientierte Kommunikationsdienste

Claudia Schmidt

Fortgeschrittene Kommunikationssysteme müssen gleichzeitig eine Vielzahl sehr verschiedener Anwendungen unterstützen, die neben dem herkömmlichen Datentransfer in zunehmendem Maße natürliche Kommunikationsformen wie Sprache und Bewegtbilder integrieren. Diese Anwendungen sind durch stark unterschiedliche Anforderungen hinsichtlich des benötigten Kommunikationsdienstes charakterisiert. Beispielsweise ist für Audio- und Videoübertragung ein kontinuierlicher Datenfluß von zentraler Bedeutung. Für einen anderen Anwendungstyp, interaktive Anwendungen, sind große Verzögerungen der Daten mit spürbaren Qualitätseinbußen verbunden. Traditionelle Kommunikationssysteme, wie z.B. der Internet-Protokollturm mit den Protokollen TCP/UDP/IP sind zur Zeit nicht in der Lage, die benötigten Dienste zu unterstützen. Eine solche Unterstützung von anwendungsorientierten Diensten erfordert neben adäquaten Mechanismen im Kommunikationssystem eine grundlegende Modifikation traditioneller Dienstschnittstellen.

Dieser Beitrag stellt zunächst die einzelnen Teilaufgaben vor, die für eine Unterstützung anwendungsorientierter Dienste zu bewältigen sind. Anschließend wird basierend auf einem Ressourcenmodell die Dienstgarantie von Anwendung-zu-Anwendung diskutiert. Die dazu benötigten Komponenten innerhalb des Kommunikationssubsystems und ihre Aufgaben sind Bestandteil des letzten Abschnitts.

1 Aufgaben bei der Unterstützung anwendungsorientierter Dienste

Kommunikationsdienste sind in der Regel durch mehrere Dienstmerkmale sowie eine spezifische *Dienstqualität* (Quality-of-Service, QoS) charakterisiert. Die ITU (International Telecommunications Union) definiert die Dienstqualität als “collective effect of service performances which determine the degree of satisfaction of a user of the service” [IT93]. Die Dienstqualität bezieht sich entsprechend dieser Definition auf die Qualität, die der Dienstanutzer an der Dienstschnittstelle angeboten bekommt. Die Bereitstellung eines Kommunikationsdienstes von Anwendung-zu-Anwendung erfordert eine Erweiterung traditioneller Dienstschnittstellen sowie eine effiziente Unterstützung des Dienstes innerhalb des Kommunikationssystems [CCG⁺93]. Dazu können die folgenden Teilaufgaben identifiziert werden [SZ94]:

- **Dienstspezifikation:** Die Dienstspezifikation befaßt sich mit der Definition des geforderten Dienstes. Innerhalb eines Kommunikationssystems ist an verschiedenen Dienstschnittstellen eine Dienstspezifikation auf einem unterschiedlichen Abstraktionsniveau erforderlich. Anwendungen bevorzugen an der *Anwendungsschnittstelle* (API) eine Beschreibung des Dienstes in einer anwendungsnahen Syntax und Semantik, die nicht mit der des Kommunikationssystems übereinstimmt. Innerhalb des Systems sind systemnahe Spezifikationen relevant. Hier wird der Dienst durch mehrere Dienstmerkmale, wie z.B. Kommunikationsform (Multicast, Punkt-zu-Punkt) und Qualitätsparameter (Durchsatz, Verzögerung, Jitter, Zuverlässigkeitsparameter) spezifiziert. Eine detaillierte Charakterisierung der Dienstqualität erfordert neben der Spezifikation von Qualitätsparametern zusätzliche Dienstklassen (deterministisch, statistisch und bestmöglich) zur Beschreibung der benötigten Garantien.
- **Dienstabbildung:** Die unterschiedlichen Dienstspezifikationen implizieren eine Abbildung des Dienstes zwischen den einzelnen Schnittstellen. Die Beschreibung des an der Anwendungsschnittstelle geforderten Dienstes muß automatisch und für die Anwendung transparent auf systeminterne Parameter abgebildet werden.
- **Aushandlung des Dienstes:** Die Aushandlung des Dienstes erfolgt in zwei Schritten. Zunächst findet eine lokale Aushandlung zwischen Anwendung und den beteiligten Komponenten im Endsystem statt. In einem weiteren Schritt, der Ende-zu-Ende-Aushandlung, werden teilweise alle Kommunikationssysteme involviert, die ein Datenstrom passiert. Hier kommen bei Diensten mit garantierter Dienstqualität spezielle Reservierungsprotokolle wie z.B. RSVP und ST-II zum Einsatz [SZ95d].
- **Erbringung des Dienstes:** Damit der geforderte Dienst erbracht wird, müssen geeignete Mechanismen zur Unterstützung der einzelnen Dienstmerkmale und Qualitätsparameter identifiziert und eingesetzt werden. In Bezug auf die Dienstqualität wird außerdem die Überwachung der aktuellen Qualitätsparameter durch einen Monitor benötigt. Sollte die erreichte Dienstqualität nicht den Anforderungen genügen, so müssen die eingesetzten Mechanismen geeignet adaptiert werden oder eine Neuaushandlung des Dienstes eingeleitet werden.

2 Ressourcenmodell

Ein Kommunikationssystem verwaltet eine Reihe von Ressourcen, die von unterschiedlichen Datenströmen gemeinsam genutzt werden. Beispiele solcher Ressourcen sind Prozessorkapazität, Speicher und Bandbreite, aber auch eher abstrakte Ressourcen wie Protokollfunktionen. Eine Möglichkeit spezifische Dienste zu garantieren, besteht in einer geeigneten Verwaltung der Kommunikationsressourcen und der Zuteilung zu den einzelnen Datenströmen basierend auf den Dienstanforderungen. Generell lassen sich in Kommunikationssystemen drei verschiedene Komponenten identifizieren, die Ressourcen auf unterschiedlichen Abstraktionsstufen verwalten (siehe Abb. 32):

- das Netz,
- das Kommunikationssystem und
- das Betriebssystem.

Jede dieser drei Komponenten kann durch ein Ressourcenmodell [SZ95c] modelliert werden. Die Verwaltung der Ressourcen ist dabei durch eine spezifische Dienstschnittstelle verborgen. Über diese Dienstschnittstelle ist es den anderen Komponenten möglich, die internen Ressourcen zu nutzen. Diese werden von Ressourcenmanagern verwaltet und zugeteilt. Die Aufgaben der Ressourcenmanager umfassen im einzelnen: Zugangskontrolltests und Reservierungen vor der Nutzung sowie die Zuteilung und Freigabe der Ressourcen [SZ95d].

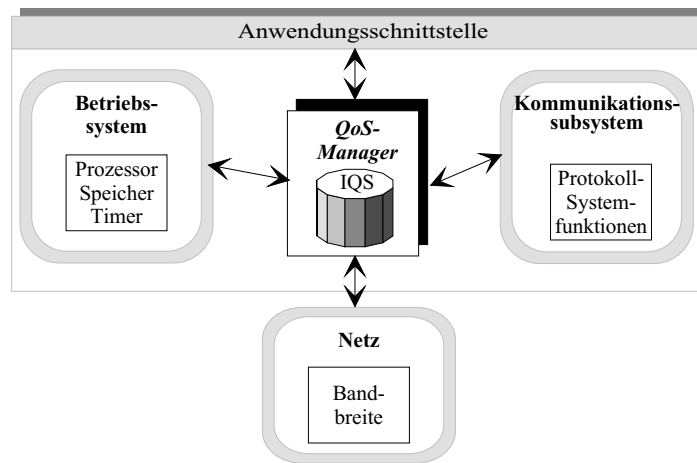


Abbildung 32. Ressourcenmodell zur Dienstunterstützung

Die *Netzkomponente* repräsentiert das Netz und den Netzzugang, z.B. die MAC-Schicht eines LANs oder die ATM-Schicht eines ATM-basierten Netzes. Sie verwaltet in jedem Zwischen- und Endsystem die Ressource Bandbreite und teilt sie den konkurrierenden Datenströmen zu. Damit die Zuteilung entsprechend des geforderten Dienstes erfolgt, sind geeignete Mechanismen zum Scheduling, zur Verkehrskontrolle und zur Staukontrolle in jedem einzelnen Kommunikationssystem einzusetzen [SZ95b].

Das *Kommunikationssystem* enthält alle kommunikationsorientierten Funktionen die nötig sind, um den Netzdienst so zu erweitern, daß der von der Anwendung gewünschte Dienst erbracht wird. Die Ressourcen sind in diesem Fall: Protokollfunktionen (z.B. Flußkontrolle, Staukontrolle, Fehlererkennung, -behebung) und Systemfunktionen (z.B. eine Protokollspeicherverwaltung). Für viele Protokoll- und Systemfunktionen existieren mehrere alternative Protokollmechanismen mit unterschiedlichem Funktionsumfang. Ein Kommunikationssystem muß in der Lage sein, basierend auf den Dienstanforderungen einen bestimmten Funktionsumfang auszuwählen, um den geforderten Dienst zu erbringen. Gleichzeitig sollte diese Menge minimal sein, daß heißt, sie sollte keine unnötigen Elemente enthalten. Dies spiegelt sich auch in aktuellen Untersuchungen wider, die zeigen, daß nicht benötigte Protokollfunktionen einen erheblichen Overhead erzeugen und damit die Leistung des Systems verringern [RSFW94].

Damit Dienste von Anwendung-zu-Anwendung garantiert werden, ist schließlich in den Endsystemen eine geeignete *Betriebssystemunterstützung* erforderlich. Das Betriebssystem verwaltet Prozessor, Speicher sowie Zeitgeber und stellt Basisdienste für Anwendungen und Kommunikationssystem zur Verfügung. Diese Betriebssystemdienste müssen im Hinblick auf eine Garantie sehr verschiedener Dienste erweitert werden. Daher sollten die Dienstanforderungen sowohl in das Scheduling der Prozessorverwaltung als auch in die Speicherverwaltung einfließen.

3 Dienstunterstützung von Anwendung-zu-Anwendung

Der geforderte Dienst zwischen Anwendungsprozessen kann nur dann garantiert werden, wenn Netz, Kommunikationssystem und Betriebssystem geeignet miteinander kooperieren. Diese Kooperation wird durch eine zentrale Komponente, den sogenannten *QoS-Manager* (siehe Abb. 32), unterstützt. Dieser regelt die Zusammenarbeit der systeminternen Komponenten und kommuniziert gleichzeitig mit der Anwendung.

Der QoS-Manager ist zunächst für die *lokale Aushandlung des Dienstes* mit der Anwendung zuständig. Die Dienstanforderung bezieht sich auf einen sogenannten Datenstrom, eine unidirektionale Verbindung. Die Dienstbeschreibung erfolgt durch eine Menge allgemeiner Dienstparameter, die für alle Anwendungen Relevanz besitzen und zusätzlich spezifische Parameter, die nur für bestimmte Anwendungsklassen von Interesse sind. Die allgemeinen Dienstparameter beschreiben den Medientyp (Daten, Video, Audio, Graphik), die Kommunikationsbeziehung (Punkt-zu-Punkt, Multicast, Multipeer) und die Zeitbeziehung der Teilnehmer (Echtzeit, interaktiv, asynchron). Spezifische Parameter für die Klasse der Videoströme betreffen beispielsweise die Frame-Rate, die Frame-Größe und das Kompressionsverfahren.

Eine weitere Aufgabe des QoS-Managers besteht in der automatischen *Hin- und Rückabbildung von Anwendungsanforderungen* in einer anwendungsnahen Syntax und Semantik auf systemrelevante Parameter. Alle systemrelevanten Parameter speichert der QoS-Manager in einer internen QoS-Struktur (IQS). Der QoS-Manager kann für die effiziente Abbildung auf eine Datenbasis zurückgreifen, in der einerseits Abbildungsregeln definiert sind und andererseits Erfahrungswerte. Damit muß eine Anwendung nicht alle Parameter spezifizieren, sondern kann für nicht spezifizierte Werte einer Standardklasse zugeordnet werden. In einem zweiten Schritt werden die Parameter der IQS auf die Parameter der systeminternen Dienstschnittstellen abgebildet. Weiterhin informiert der QoS-Manager die anderen Komponenten, falls ein neuer Datenstrom aktiv wird oder falls sich die Charakteristiken eines existierenden Datenstroms ändern.

Alle systeminternen Komponenten sind in die *Aushandlung* und in die *Erbringung des Dienstes* involviert. Bevor ein neuer Datenstrom aktiv wird, prüfen Ressourcenmanager, ob der Dienst unterstützt werden kann und reservieren die entsprechenden Ressourcen. Betrachtet man die Qualitätsparameter Durchsatz, Verzögerung, Jitter und Zuverlässigkeit, so ist eine Unterstützung der einzelnen Parameter von jeweils unterschiedlichen Komponenten abhängig. Traditionelle Kommunikationssysteme beschränken sich auf die Unterstützung von Zuverlässigkeitsparametern, wie beispielsweise die Behandlung von Duplikaten und Verlusten. Dazu wird die Fehlerkontrolle des Netzes durch eine Vielzahl von Fehlererkennungs- und Fehlerbehebungsmechanismen innerhalb des Kommunikationssubsystems verbessert. Die Unterstützung der leistungsorientierten Parameter Durchsatz, Verzögerung und Jitter gestaltet sich merklich schwieriger und wird von traditionellen Systemen nicht übernommen. Diese Parameter werden von Protokollfunktionen wie Fluß- und Ratenkontrolle unterstützt, aber gleichzeitig sind sie auch direkt abhängig vom Betriebssystem und jedem Zwischensystem im Netz. So wird die Verzögerung stark von den Schedulingmechanismen in den Zwischensystemen beeinflusst. Um leistungsorientierte Qualitätsparameter zu garantieren, müssen Betriebssystem, Netzwerk und Kommunikationssystem miteinander kooperieren und die Ressourcenverwaltung aufeinander abstimmen.

4 Dienstspezifische Kommunikation

Moderne Kommunikationssysteme sollten in der Lage sein, einer Vielzahl von Anwendungen gleichzeitig Dienste anzubieten [CSS94]. Dazu sind anwendungsspezifische Protokollkonfigurationen zu erstellen, die den Dienst unterstützen, aber keinen unnötigen Protokolloverhead und somit keine Leistungsminderung erzeugen [SSZ94]. Diese Aufgabe wird durch einen *Protokoll-Manager* (siehe Abb. 33) wahrgenommen. Bei der Etablierung eines neuen Datenstromes informiert der QoS-Manager den Protokoll-Manager über die Dienstanforderungen. Der Protokoll-Manager nutzt eine Datenbasis mit Regeln zur Ermittlung der anwendungsspezifischen Protokollfunktionalität. Dazu selektiert er in einem ersten Schritt die benötigten Protokollfunktionen. Sollten mehrere alternative Protokollmechanismen für eine Protokollfunktion existieren, so wählt er in einem zweiten Schritt einen geeigneten Protokollmechanismus aus. Schließlich berechnet er basierend auf der Dienstcharakterisierung die initialen Konfigurationsparameter für die selektierten Protokollmechanismen.

Die Datenbasis besteht aus einfachen Regeln, die Wechselwirkungen zwischen Protokollfunktionen und -mechanismen einerseits und den systemspezifischen Dienstanforderungen andererseits beschreiben. Basierend auf diesen Regeln ist der Protokoll-Manager in der Lage, spezifisch für jede Anwendung die benötigten Protokollfunktionen zu selektieren. Damit nicht benötigte Funktionen keinen Overhead

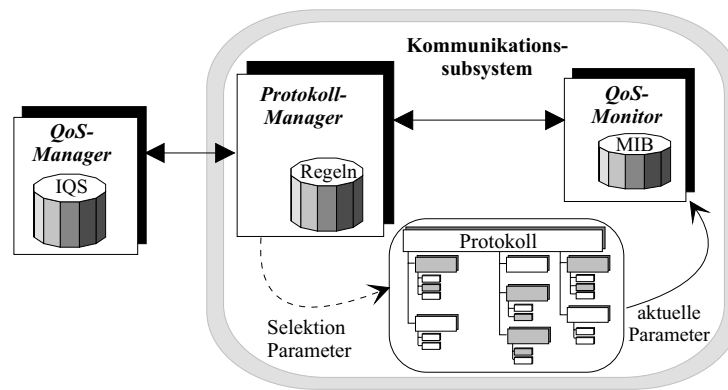


Abbildung 33. Dienstunterstützung im Kommunikationssystem

erzeugen, muß das Kommunikationsprotokoll jedoch modular aus einzelnen Funktionsblöcken aufgebaut sein. Einfache Regel der Datenbasis sind:

```

if (throughput guarantee | delay guarantee)
then select connection with resource reservation
if (deterministic reliability guarantee)
then select retransmission-based scheme
if (application_data_size > network_data_size)
then select segmentation/reassembly
if (media_type == (audio | video))
then select isochronous delivery

```

Während der Datentransferphase überwacht ein *QoS-Monitor* die momentan erreichte Dienstqualität. Dazu berechnet er basierend auf aktuellen Protokollparametern die resultierenden Qualitätsparameter und speichert diese in einer QoS-MIB (Management Information Base). Verletzungen der ausgehandelten Dienstqualität werden durch einen Vergleich der aktuellen Werte mit Grenzwerten festgestellt. Leichtere Verletzungen der Dienstqualität können oftmals durch eine Adaption systeminterner Parameter behoben werden, bei signifikanten Verletzungen muß eine Neuaushandlung der Dienstqualität stattfinden.

5 Aktueller Stand

In bisherigen Arbeiten wurde zunächst eine Umgebung für das beschriebene Modell geschaffen. Grundlage dabei bildet eine Portierung des am Institut entwickelten Transportsystems PATROCLOS [BS94] unter das Betriebssystem OSF/1. Dabei wurde eine geeignete Prozeßstruktur gewählt, um dienstorientierte Betriebssystemfunktionen, wie z.B. Scheduling basierend auf Qualitätsparametern zu unterstützen. Parallel dazu wurden die Einflüsse von Schedulingalgorithmen auf Qualitätsparameter und verschiedene Scheduling-Architekturen untersucht und ein ausgewähltes Verfahren implementiert. Aktuelle Arbeiten beschäftigen sich mit der Realisierung der Anwendungsschnittstelle und der Abbildung einer anwendungsorientierten Dienstbeschreibung auf systemrelevante Parameter. Gleichzeitig wird ein allgemeines Monitoringmodell unter Berücksichtigung des Netzwerkmanagementprotokolls SNMP entwickelt und implementiert. Weiterhin sind die Entwicklung des QoS-Managers und des Protokoll-Managers mit den entsprechenden Datenstrukturen geplant.

Integration des Netzwerkmanagements in das Management verteilter Anwendungen

Jochen Seitz

1 Ausgangspunkt

Der große Bereich „Netzwerkmanagement“ kann einige umfangreiche Standards aufweisen, die in zahllosen Produkten Einsatz finden. Sei es das für die Verwaltung des Internets definierte *Simple Network Management Protocol (SNMP)*, das in zwei (inkompatiblen) Versionen vorliegt, sei es der *Common Management Information Service (CMIS)*, wie er von der internationalen Standardisierungsorganisation definiert wurde, oder neuerdings Objektmanagementansätze, wie sie beispielsweise mit der *Common Object Request Broker Architecture CORBA* begründet werden, all diesen Managementansätzen ist eine zentrale Architektur gemein (siehe Abbildung 34): Das Netzwerk respektive verteilte System wird von einem zentralen Punkt aus verwaltet. Der momentane Trend hin zu Managementplattformen [Jan94], z.B. das *Distributed Management Environment DME* [Sei94c] der Open Software Foundation, verstärkt den Zentralisierungsaspekt noch beträchtlich.

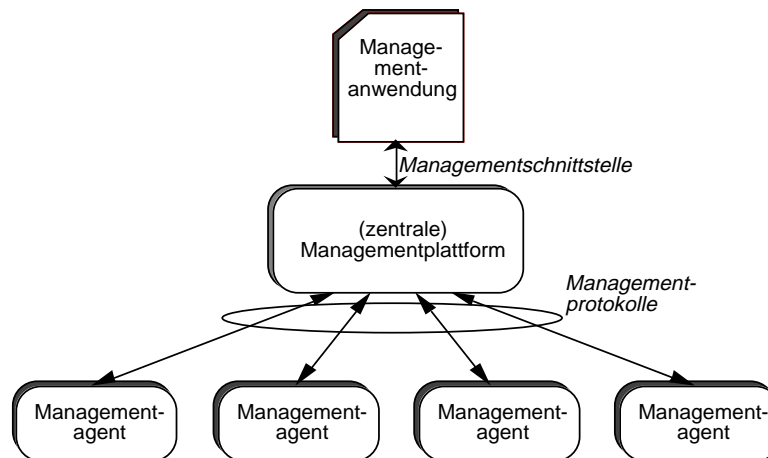


Abbildung 34. Herkömmlicher, zentraler Managementansatz

Diese Zentralisierung ist bei der Verwaltung von Netzwerken durchaus wünschenswert, da hierbei alle Knoten von einer Stelle, sprich von einem Terminal aus verwaltet werden können. Doch schon mit zunehmender Größe des zu verwaltenden Netzes muß, um eine gewisse Übersichtlichkeit zu wahren, eine hierarchische Verwaltung ins Leben gerufen werden, bei der bestimmte Teile eines großen Netzes lokal verwaltet werden und erst bei lokal nicht mehr lösbaren Problemen oder übergreifenden Planungsaufgaben eine übergeordnete Managementstufe eingeschaltet wird.

Doch auch diese Vorgehensweise reicht bei der Verwaltung eines verteilten, auf autonomen Komponenten, den sogenannten *Agenten*⁸, basierenden Systems nicht aus. Ziel eines solchen Systems ist nämlich, daß es sich selbst verwalten soll und daß somit die einzelnen Komponenten alle die Fähigkeit besitzen sollen, einerseits Managementanfragen beantworten zu können, andererseits selbst solche Anfragen abzusetzen. Die dazu notwendige Managementgrundfunktionalität soll im vorliegenden Beitrag definiert und am Beispiel beschrieben werden.

Dazu wird in Kapitel 2 vorgestellt, welche unterschiedlichen Managementbereiche für die Verwaltung eines beschriebenen Agentensystems differenziert werden können. Kapitel 3 detailliert einen Bereich davon, der sich mit der Eingliederung von Netzwerkmanagementfunktionalität in die einzelnen Agenten befaßt. Zur Verdeutlichung der Relevanz dieses Themas führt Kapitel 4 ein Beispielszenario an. Kapitel 5 schließt den Beitrag mit einem zusammenfassenden Ausblick.

⁸ Der Ausdruck „Agent“ sollte allerdings *nicht* mit dem Begriff des Managementagenten verwechselt werden!

2 Differenzierung des Managements von Agentensystemen

Ein Agentensystem besteht, wie bereits erwähnt, aus einer Menge kooperierender autonomer Instanzen, den Agenten. Diese Agenten sollen ergo gemeinsam eine bestimmte Aufgabe erfüllen, wobei jeder Agent für sich funktionstüchtig ist und entsprechend einen bestimmten Teil der Aufgabe zu bewerkstelligen hat. Aus dieser — sehr allgemeinen — Definition eines Agentensystems lassen sich grob drei unterschiedliche Managementbereiche klassifizieren:

1. *Management lokal zum Agenten*

Jeder Agent muß sich selbst mit Information über die Ablaufumgebung versorgen können, damit er seinen Bedürfnissen entsprechend Netzwerk- und Rechnerressourcen in Anspruch nehmen kann. Der Austausch von Managementinformation findet daher zwischen Agent und dem Rechnersystem bzw. Netzwerk statt, wobei Managementagenten (in der Abbildung als „MA“ bezeichnet) adressiert werden (Abbildung 35, Punkt 1).

2. *Management unter den Agenten*

Zum autonomen Ablauf der Agenten innerhalb eines Systems ist es unbedingt notwendig, daß die Agenten untereinander ebenfalls Verwaltungsinformation austauschen. Hierunter fallen z.B. das Anbieten gewisser Dienste oder das Lokalisieren von Agenten (Abbildung 35, Punkt 2).

3. *zentralisiertes Management des Agentensystems*

Schließlich ist auch für ein Agentensystem — zumindest zur Überwachung oder zum Debugging — auch eine zentrale Instanz möglich, die mit jedem Agenten des System Managementinformation austauschen kann (Abbildung 35, Punkt 3).

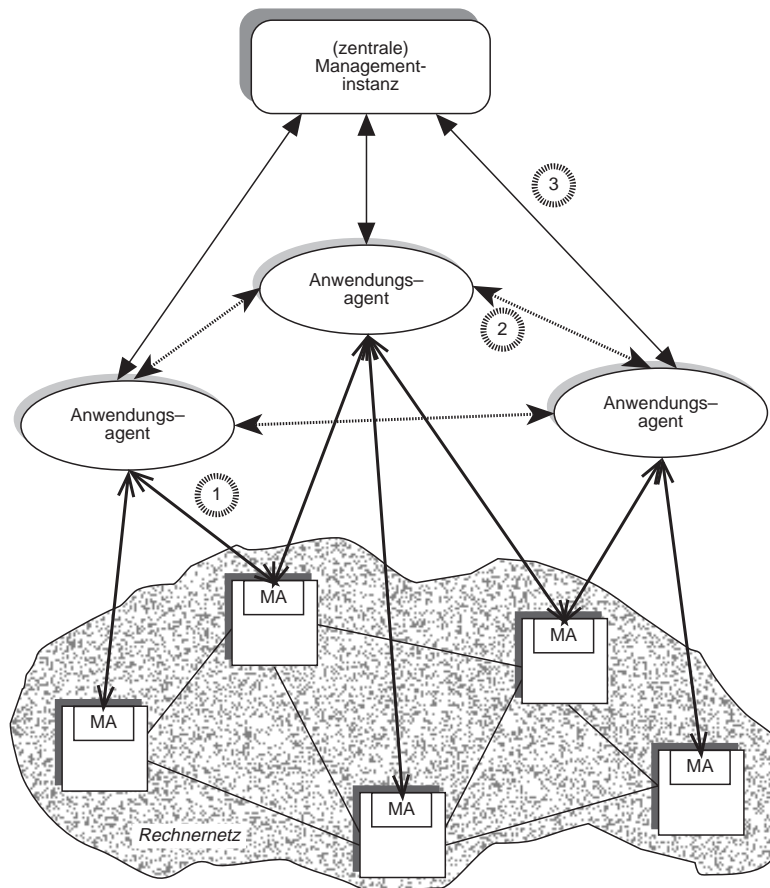


Abbildung 35. Management in einem Agentensystem

Von diesen drei Bereichen sind die letzten beiden noch am wenigsten reglementiert. Sowohl die bereitzustellende Managementinformation als auch das zum Transport derselbigen verwendete Managementprotokoll sind frei wählbar und auf das jeweilige Anwendungsproblem anpaßbar. Der erste Punkt allerdings sollte auf bereits existierende Gegebenheiten Rücksicht nehmen, die mit der Verwaltbarkeit des

zugrundeliegenden Rechnersystems gegeben sind. Daher wird hierbei der Agent (des Agentensystems) mit den Managementagenten über das von diesen unterstützte Managementprotokoll (also beispielsweise das *Simple Network Management Protocol (SNMP)* oder das *Common Management Information Protocol (CMIP)*) die dort vorhandene Managementinformation abfragen beziehungsweise von diesen Managementagenten Informationen über sich kritisch verändernde Parameter erhalten. Dieser Punkt soll im folgenden Kapitel detaillierter untersucht werden.

3 Integration von System- und Netzwerkmanagementfunktionalität in einen Anwendungsagenten

Die einem Agentensystem zugrundeliegende Rechner- und Netzwerktopologie bietet bereits die Möglichkeit, über ein Managementprotokoll bestimmte Verwaltungsdaten, wie z.B. die Anzahl der auf der Transportschicht gesendeten Pakete abzufragen. Dies wird durch Managementagenten realisiert, die auf den einzelnen Rechnern als Demon-Prozesse laufen. Managementagenten sind beliebig erweiterbar, so daß sie beispielsweise auch Informationen zum Systemzustand liefern können, also beispielsweise die momentane Anzahl der Benutzer oder die derzeitige CPU-Auslastung.

Leider hat sich, was das Netzwerkmanagement angeht, bis dato noch kein Standard durchgesetzt und es ist auch nicht abzusehen, wann dies der Fall sein wird. Daher muß gerade in einem heterogenen Netz damit gerechnet werden, daß nicht nur ein Managementprotokoll – und damit verbunden nicht nur ein Managementmodell – existiert, sondern daß wie bereits erwähnt verschiedene Standards realisiert wurden. Um jetzt von allen Managementagenten Informationen erhalten zu können, müßte jeder Anwendungsagent alle eingesetzten Managementstandards implementiert haben. Dies würde jedoch bedeuten, daß zum einen über mehrere unterschiedliche Protokolltürme verfügt werden müßte und daß zum anderen der Umgang mit Managementinformation für jeden Standard getrennt behandelt werden müßte. Ergebnis wären Agenten, die zum Großteil aus Code für das Management bestünden und auch hauptsächlich mit Managementaufgaben beschäftigt wären.

Somit sollten die Managementfunktionalität in den Agenten auf ein Minimum beschränkt und die restliche benötigte Funktionalität auf andere Instanzen ausgelagert werden. Dazu bietet sich eine Abstraktion der real im Netz verfügbaren Rechner durch sogenannte *logische Knoten* an (Abbildung 36, [ZS93]). In diesen logischen Knoten wird das in [Sei94a, Sei94g] beschriebene *allgemeine Management Application Programming Interface AM-API* realisiert, mit dem der logische Knoten generische Managementdienste anbietet und geforderte Dienste in konkrete, auf dem jeweils erforderlichen Managementstandard basierende Managementaktionen umsetzt.

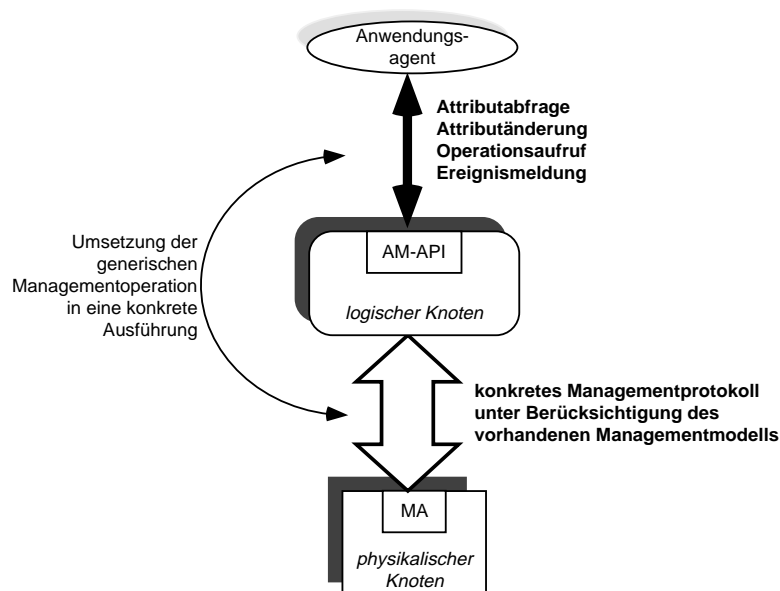


Abbildung 36. Einführung von sogenannten „logischen“ Knoten

Anwendungsagenten können über diese API folgende Dienste in Anspruch nehmen:

1. Mit der bestätigten *Attributabfrage* können vom zugrundeliegenden realen Rechnernetz Managementinformationen abgefragt werden.
2. Über eine ebenfalls bestätigte *Attributänderung* sind Modifikationen an den realen Ressourcen durch Anwendungsagenten möglich.
3. Falls die Managementeinheiten im realen Rechnernetz zusätzliche Operationen zur Verfügung stellen, können diese über einen bestätigten *Operationsaufruf* vom Anwendungsagenten in Anspruch genommen werden.
4. Letztendlich kann der logische Knoten bei Ausnahmesituationen über unbestätigte *Ereignismeldungen* den Anwendungsagenten davon informieren.

Somit haben alle Anwendungsagenten die gleiche Managementschnittstelle zur Verfügung, die den Zugriff auf unterschiedliche Managementprotokolle und -modelle unter Zuhilfenahme von logischen Knoten transparent verdeckt. Dieses Vorgehen ist durchaus legitim, da einerseits die logischen Knoten benötigt werden, um ein weitestgehend flexibles Agentennetz zu realisieren, und da andererseits die Anzahl der logischen Knoten nie größer als die Anzahl der Anwendungsagenten ist, sondern im allgemeinen immer wesentlich kleiner, so daß dadurch der Managementoverhead reduziert werden kann.

4 Ein illustrierendes Beispiel

Anhand eines kleinen Beispiels soll das Zusammenspiel zwischen Anwendungsagenten, logischem und physikalischem Knoten illustriert werden. Gegeben sei ein Anwendungsagent als Rechenprozeß, welcher frei über die Maschine, d.h. den physikalischen Knoten, entscheiden kann, auf der er läuft. Ist die Maschine zu stark belastet, wird er sich auf eine andere Maschine migrieren. Als relevantes Merkmal für die Auslastung einer Maschine soll der Einfachheit halber die Anzahl der Benutzer gesehen werden, die derzeit auf der Maschine angemeldet sind. Um festzustellen, ob eine Maschine geeignet ist, als Ablaufumgebung für den Anwendungsagenten zu dienen, schickt der Agent eine Attributabfrage mit dem Parameter Number of Users an den die Maschine repräsentierenden logischen Knoten.

Dieser muß die generische Abfrage nach der Benutzerzahl umsetzen. Wenn beispielsweise der Managementagent auf der Maschine die „Host Resources MIB“ [GW93] verwaltet, könnte dies mit einem SNMP-GetRequest nach dem Managed Object hrSystemNumUsers geschehen. Basiert der Managementagent auf dem ISO/OSI-Managementrahmenwerk, wird die Abfrage in ein M-Get-Request umgesetzt, wobei als Managed Object beispielsweise eine Instanz des UxObj1 adressiert werden kann, welche als Attribut nUsers besitzt [Ucl92]. Besitzt der physikalische Knoten überhaupt keinen Managementagenten, könnte der logische Knoten die Information unter Zuhilfenahme von Unix-Systemaufrufen wie z.B. einem who die gewünschte Information dennoch bestimmen. Der Anwendungsagent wird in jedem Fall eine Integerzahl als Antwort erhalten, ohne zu wissen, auf welche Art und Weise diese Information gewonnen wurde.

Eine weitere Möglichkeit im vorliegenden Fall wäre, daß die Managementinformation überwacht und bei Überschreiten einer gewissen Schranke (z.B. Number of Users > 7) der Anwendungsagent davon informiert wird. Insgesamt kann man erkennen, daß dieser nur über wenig Intelligenz für das Management verfügen muß, wohingegen der logische Knoten die fehlende Intelligenz bereitstellen wird.

5 Zusammenfassung und Ausblick

Die Verwaltung von verteilten Systemen wird mit der Entwicklung hin zu Agentennetzen immer mehr in den Vordergrund treten. Daher ist es wichtig, bereits frühzeitig über Mechanismen verfügen zu können, die beim Management eines solchen Netzes helfen können. Dabei kann man, zumindest was die autonome Verwaltung der einzelnen Agenten angeht, auf die Ergebnisse zurückgreifen, die im Netzwerkmanagement bereits erzielt wurden.

Um ein Agentennetz allerdings von einem zu großen Management-Overhead zu bewahren, sollte die für die Verwaltung benötigte „Intelligenz“ nicht in jedem Anwendungsagent zu finden sein, sondern sie sollte den Agenten über logische Knoten zur Verfügung gestellt werden. Diese logischen Knoten bieten in Richtung Anwendungsagenten das allgemeine Management-API an und setzen ankommende generisch gehaltene Anfragen in Managementaktionen um, die auf einem konkreten Managementprotokoll mit dazugehörendem Managementmodell basieren.

Die generelle Verwaltung eines Agentennetzes, sei es durch Managementkommunikation zwischen den Agenten oder durch eine (zentrale) Managementinstanz, wird durch den vorgeschlagenen Ansatz allerdings nicht reglementiert. Hier sind eigene Lösungsmöglichkeiten zu entwickeln, die jedoch mit der Zunahmen von Agentennetzen immer dringender werden.

Unterstützung von Kooperation in verteilten Systemen

Jörg Sievert

1 Einleitung

Im Anwendungsbereich verteilter Systeme ist Kooperation ein naheliegendes Thema. Es manifestiert sich in vielen Arbeitsbereichen, insbesondere aber in der Unterstützung von Kooperation zwischen Menschen. Um Kooperation zwischen Menschen zu unterstützen, muß untersucht werden, wie Menschen in Organisationen zusammenarbeiten. Dazu werden einige Forschungsgebiete betrachtet, deren Erkenntnisse zu diesem interdisziplinären Forschungsgebiet beitragen können [Pow91].

2 Kooperation

Die Informatik steht in diesem Gebiet nicht allein auf weiter Flur, sondern trifft auf verwandte Forschungsgebiete. Die *Organisationspsychologie* beschäftigt sich mit dem Verhalten von Menschen in Organisationen. Im modernen *Management* wird der Einfluß von Kooperation innerhalb von Unternehmensbereichen und zwischen Unternehmen seit Jahren als positiv gepredigt. Die *Linguistik* betrachtet besonders die Analyse der Benutzung von Sprache durch den Menschen, insbesondere in Gruppen. In sich selbst ein interdisziplinäres Gebiet, konzentriert sich die *Verteilte Künstliche Intelligenz* auf die Zusammenarbeit unabhängiger Systeme und insbesondere von Wissensbasen.

2.1 Organisationspsychologie

Im allgemeinen Gebiet der Psychologie versteht sich die Organisationspsychologie als angewandte, empirische Psychologie [vR92][Pow91][Gro94] [Wei87][Gre83], im Gegensatz zur theoretischen oder praktischen Psychologie. Im folgenden werden Gruppenentscheidungen, Verhandlungen und Besprechungen und Konferenzen als spezielle Themen der Organisationspsychologie eingeführt, die für Gruppenarbeit eine besondere Bedeutung haben.

Gruppenentscheidungen: Das Verhalten von Menschen in Gruppen ist ein zentraler Forschungspunkt in den aktuellen Sozialwissenschaften. Gruppen nehmen im Laufe der Zeit eine eigene Kultur an. Sie entwickeln Verhaltensnormen, die teilweise durch Erfahrungen geprägt sind und teilweise von Mitgliedern in die Gruppe eingebracht werden. So entwickelt sich auch ein Rollenverhalten in der Gruppe. Frühe Studien teilten das Wirken von Menschen in Gruppen auf in *Aufgabenspezialisten*, welche die vorliegende Arbeit durchführen, und in *Sozialspezialisten*, welche zum Gruppenzusammenhalt beitragen. Innerhalb einer Gruppe orientieren sich Menschen häufig an den Perspektiven der Gruppe, auch wenn diese sehr unterschiedlich von den persönlichen Ansichten sind. Gruppenentscheidungen tendieren dazu, extremer oder risikoreicher zu sein als Entscheidungen durch einen Einzelnen. Dieses Verhalten wird auch *Groupthink* genannt. Die folgenden Symptome sind typisch: Ein Übertriebener Optimismus führt zu extremen Risiken und Ignoranz gegenüber offensichtlichen Gefahren. Aussagen gegen die akzeptierte Gruppenmeinung werden diskreditiert. Die Position der Gruppe wird als ethisch angesehen, da niemand sie in Frage stellt. Bedingt durch ihren Optimismus lehnt die Gruppe andere Meinungen pauschal ab. Mitglieder der Gruppe zensieren eigene, unangepaßte Meinungen. Durch die Unterdrückung widersprüchlicher Meinungen wird ein vermeintliches Mehrheitsverhältnis gebildet.

Verhandlungen: Im Gegensatz zu Gruppenentscheidungen, in denen Gruppen zusammenarbeiten, um gemeinsam zu einer Entscheidung zu kommen, stellen Verhandlungen eher den Konflikt in Gruppen dar. Verhandlungen sind Gruppeninteraktionen, in denen Teilnehmer einen Konsens zu Themen finden, in denen sie sich kognitiv und politisch unterscheiden. Auf der kognitiven Seite müssen die Teilnehmer die (fremden) Konzepte untereinander verstehen, während auf der politischen Seite die Motive, Stärken und Schwächen der Teilnehmer im Mittelpunkt stehen. In Verhandlungen kann ein *Mediator* eine kritische Rolle spielen. Als unparteiische Instanz klärt er die Positionen und sucht Wege zum Konsens. Hier sind eine Reihe von Techniken bekannt wie z.B. *Konzessionsaustausch*: Die Teilnehmer tauschen Konzessionen aus, um schrittweise ihre Bedingungen zu erfüllen. Es existieren einige Versuche, diese Techniken

in Expertensystemen zu verwenden, die selbständig Verhandlungen führen sollen. Die Erfahrungen zeigen, daß sehr viel Domänenwissen notwendig ist, um dieses erfolgreich zu tun. Ein realistischerer Ansatz ist es, den Mediator bei seiner Arbeit zu unterstützen. Ein Hypertextsystem könnte die Positionen der Teilnehmer sichtbar machen. Sobald eine Position derart definiert ist, können andere Teilnehmer und der Mediator Kommentare und Fragen an diese Positionen hängen. Der Mediator muß dann in der Lage sein, Teile der Positionen zu Vorschlägen zusammenzufassen. Sehr wesentlich ist auch die Darstellung, wie es zu einer Entscheidung gekommen ist, damit die Situation für alle Teilnehmer nachvollziehbar ist.

Besprechungen und Konferenzen: Eine Vielzahl von Studien hat gezeigt, daß die wichtigste Aktivität in kooperativem Arbeiten die Kommunikation ist. Im Bereich der Softwareentwicklung fanden Studien, daß Entwickler bis zu 80% ihrer Zeit damit verbrachten, mit ihren Kollegen zu interagieren, während sie nur 20% am Computer verbrachten. Speziell Besprechungen können nach ihrem Zweck, ihrer Häufigkeit, den Anwesenden und dem Protokoll charakterisiert werden. Eine der Hauptaufgaben bei einer Besprechung ist die Festlegung des Ziels der Besprechung; obwohl scheinbar elementar, wird gerade dieser Schritt häufig vernachlässigt. Häufige Ziele sind Informationen austauschen, Aktivitäten koordinieren, Entscheidungen treffen, Meinungen kennenlernen, Lösungen vorschlagen und ganz einfach kreatives Denken zu fördern. Weitere Elemente einer Besprechung sind die Agenda, der Leiter und der Sekretär einer Besprechung. Diese Elemente sollte ebenso wie das Ziel schon am Beginn der Besprechung festgelegt werden. Die Agenda sollte dabei das Ziel erwähnen, und eine Liste der zu besprechenden Punkte enthalten, wenn möglich mit der vorgegebenen Zeit. Nach der Besprechung müssen die Ergebnisse zusammengefaßt und an alle Teilnehmer verteilt werden. Gegebenenfalls muß ein Bericht verfaßt werden. Vor allem das Verhalten der Leiter wurde erforscht. Es haben sich zwei Stilrichtungen kristallisiert. *Teilnehmende* Leiter werden von Teilnehmern häufig bevorzugt, sie führen aber zu einer langsameren Entscheidungsfindung. *Autokratische* Führer kommen schneller zu Entscheidungen, nutzen aber nicht das Potential ihrer Mitarbeiter und reduzieren sogar deren Zufriedenheit. Ein Leiter sollte beide Stilrichtungen vereinen und der Situation angepaßt anwenden können. Soweit möglich, sollte ein Leiter objektiv sein und die Besprechung durch kurze Zusammenfassungen und das Ansprechen von Teilnehmern kontrollieren. Er muß zwar kein Experte im besprochenen Gebiet sein, aber zumindest Interesse am Thema haben und wissen, wann die Besprechung beendet werden muß. Nur bedingt sollte der Entscheidungsträger auch der Leiter einer Besprechung sein sollte. In vielen Fällen ist es sinnvoll, die Rollen zu trennen und als Leiter eine Person einzusetzen, die in der Methodik geschult ist.

2.2 Management

Nach *Fayol* (1916) ist Koordination eine zentrale Managementfunktion und nach *K. Marz* neben der Herrschaftssicherung die zentrale Funktion des Managements überhaupt [Sta87]. Dabei betrachtet das Management, als Wirtschaftswissenschaft, den Einfluß von Organisationsstrukturen auf Koordination und Motivation in Unternehmen. Bekannte Organisationsstrukturen sind die *Regionalorganisation*, *Divisionalorganisation* und die *Funktionalorganisation*. Die 70er Jahre brachten eine Vielzahl von weiteren organisatorischen Gestaltungsansätzen. Typisch für den Eifer der Zeit sind die Management-by-Ansätze, z.B.: Management by ideas, objectives, breakthrough, crisis, motivation, delegation, exception, results, systems und direction and control. In dieser Zeit waren auch gerade Ansätze *kooperativer* Führung im Gespräch, welche die *autoritäre* Führung ablösen sollten. Begleitet wurden diese Ansätze von Kreativitätstechniken, deren Durchführung manchmal lebhaft an Kinderspiele erinnert [Hau22]: das Rollenspiel, das Utopienspiel, die Advocatus-diaboli-Methode, das wohlbekannte Brainstorming, die Methode 6.3.5, die CNB-Methode, die Synektik, die Morphologische Analyse und die Szenariotechnik.

Die Aufgaben der Koordination sieht Gemünden [Gem90] in der Koordination von *Zielen*, *Ressourcen* und *Informationen*. Dabei unterliegt die Koordination gewissen Interdependenzen. Diese sind *gepoolte Interdependenzen*, welche konkurrierenden Zugriff darstellen, *sequentielle Interdependenzen*, also die Bearbeitung eines Objekts in Reihenfolge durch verschiedene Bereiche und *reziproke Interdependenzen*, einer Erweiterung der sequentiellen Interdependenzen durch die Rückkehr eines Objekts zu einer schon besuchten Instanz. Im gleichen Text nennt Gemünden einige Techniken zur Koordination in Unternehmen. *Verrechnungspreise* werden innerhalb von Unternehmen benutzt, um die Interaktion zwischen Unternehmensbereichen kontrollieren zu können. Die Vorteile liegen in der Marktnähe der Technik und der erfolgten Motivation. *Programme* sind hier die Umsetzung des entsprechenden Begriffs aus der Informatik. Ein Programm stellt eine Menge von Aktivitäten dar, die inhaltlich bestimmt und verknüpft sind und in einer bestimmten Reihenfolge ablaufen. Die Vorteile dieser Technik liegen in der sicheren Durchführung von vorbestimmten, statischen Abläufen. Delegation ist die Übertragung von Kompetenzen. *Konsequente Delegation* ist die möglichst vollständige Definition und überschneidungsfreie Ordnung

von Aufgabenbereichen zwischen Vorgesetztem und Untergebenem. Für nicht erfaßte Aufgaben gilt die Substitutive Allkompetenz des Vorgesetzten, für diese Aufgaben ist also prinzipiell der Vorgesetzte verantwortlich. Wesentlich ist, daß der Vorgesetzte nicht in den Entscheidungsspielraum des Untergebenen eingreift, und daß der Untergebene seine Entscheidungen durch Zeichnung vertritt. Eine Vielzahl von Prozessen im betrieblichen Bereich lassen sich in Teilprozesse zerlegen. Die *Ablauforganisation* fügt diese Teilprozesse wieder in eine Reihenfolge zusammen. Ihr Ziel ist es, Parallelarbeit zu ermöglichen und Wartezeiten im Prozeß zu reduzieren. Sie kann dazu Einfluß auf die Anfangstermine und Beendigungstermine bzw. auf die Geschwindigkeit von Prozessen nehmen.

2.3 Linguistik

Die Linguistik selbst teilt sich in eine Vielzahl von Bereichen auf. Die *Pragmatik* analysiert die Benutzung der Sprache, im Gegensatz zu ihrer Struktur (Syntax) und Bedeutung (Semantik). Sie ist die Grundlage für die Analyse der sozialen Abläufe in Konversationen zwischen Menschen.

Sprechakttheorie: Eine Äußerung in einer Konversation zwischen Menschen kann, wie von vielen leicht nachvollziehbar, mehrere Bedeutungen tragen. Insbesondere können Äußerungen ein soziales Ereignis darstellen. Dieses Konzept ist wesentlich, da die Forschung aus einer Analyse von Gesprächen versucht, grundlegende soziale Elemente einer Unterhaltung zu finden. Searle untersuchte 1969 Bedingungen für die Ausführung von Unterhaltungen. Er fand drei Typen: *Inhalt (content)* der Unterhaltung, *Vorbereitung (preparation)* für die Unterhaltung und *Aufrichtigkeit (sincerity)* der Sprecher. Danach gruppierte Searle derartige Unterhaltungen in 5 Kategorien:

1. *bindend (assertive)*, ein Sprecher bindet sich an eine Aussage
2. *leitend (directive)*, Überzeugung eines Zuhörers
3. *zusichernd (commissive)*, ein Sprecher bindet sich an eine zukünftige Aktion
4. *deklarativ (declaration)*, aus einer Proposition wird ein Axiom
5. *expressiv (expressive)*, drückt einen psychologischen Zustand aus

Diese Kategorien und die vorher gefundenen Bedingungen lassen die Folgerung zu, daß Aussagen in Unterhaltungen nur in Referenz zur sozialen Situation interpretiert werden können, wie sie von den Teilnehmern empfunden wird. In wenigen speziellen Situationen wird diese Situation externalisiert, wie z.B. bei legalen Dokumenten; im allgemeinen ist die Situation nur in den Köpfen der Teilnehmer und insbesondere verändern die Aussagen mentale Objekte wie Überzeugungen, Absichten, Rechte und Verantwortlichkeiten. Diese Einsichten haben zu Annahme geführt, daß sich einzelne Sprechakte zu funktionalen Sequenzen gruppieren. Beispiele dafür sind *Frage-Antwort* und *Vorschlag-Antwort* Paare. Mit Hilfe dieser Sequenzen entwickelten Levin und Moore das *Dialogue Game*, eine soziale Routine, in der Agenten gemeinsame Ziele erreichen. Der bekannteste Ansatz, der aus diesen Anfängen entstand, ist das Transitionsnetz von Winograd und Flores [WF86], in dem Pfade den möglichen Verlauf einer Konversation darstellen.

Konversationsanalyse: Parallel zu der Entwicklung der Sprechakttheorie untersuchten Soziologen scheinbar präzise, aber undefinierte Regeln, wie menschliches Verhalten in einer Kultur interpretiert wird. Zum Beispiel fand man in Telefongesprächen Regularitäten, wie ein Gespräch eröffnet und beendet wird. Die Eröffnung und Beendigung eines Gesprächs ist besonders interessant, da hier die Konversation selbst reguliert wird, während der Inhalt der Konversation noch unberührt ist. Die beobachteten wiederkehrenden Muster in Konversationen dienen dabei elementaren Zielen, wie sicherzustellen, daß alle relevanten Themen besprochen wurden oder Einigung zu erzielen, daß die Konversation enden sollte. Eine weitere sehr wesentliche Erkenntnis betrifft den Wechsel zwischen Sprechern in einer synchronen Konversation (engl.: floor control). Ein typisches Verhalten ist die explizite Weitergabe des Rederechts durch den Redner oder eine weitere Person. Ebenso bekannt ist die Übernahme des Rederechts durch entsprechend forschendes Auftreten.

Bei seinen Studien zu Gesprächseröffnungen fand Schegloff, daß Konversationen zwischen zwei Personen typischerweise mit einem Aufruf zur Kommunikation anfängt, der durch eine Bestätigung beantwortet wird. Es folgt ein kurzer Austausch persönlicher Informationen und eine Nachfrage, ob der Gesprächspartner Zeit hat. Das letztere wird am Ende des Gesprächs zur Beendigung verwendet. Dann wenden sich beide dem eigentlichen Grund des Gesprächs zu.

Bei ihren Studien zu Gesprächsbeendigungen fanden Schegloff und Sacks, daß Konversationen zwischen zwei Personen typischerweise mit einem Austausch von idiomatischen Ausdrücken enden, welche

von beiden Personen erbracht werden, und zwar zeitlich nebeneinander und teilweise überlappend. Falls dieser Austausch unterbrochen wird, z.B. weil eine Person ein weiteres Thema einbringt, dann kann der Austausch später nicht fortgeführt werden, sondern muß erneut angefangen werden.

Diese Erkenntnisse sind teilweise direkt in Arbeiten an CSCW-Systemen geflossen, um neue Kommunikationssysteme mit elementaren Koordinationsmitteln zu erweitern. Aus der Konversationsanalyse resultieren Erkenntnisse, wie: Bei Gesprächen muß klar sein, wer teilnimmt, welches Thema diskutiert wird, ob es das Hauptthema oder nur ein Seitenzweig ist, wer gerade das Rederecht hat, wann ein Thema beendet wird und wann das Gespräch beendet wird. Bei asynchronen Konferenzen treffen einige Konzepte zwar nicht mehr ganz so zu, wie z.B. die Weitergabe des Rederechts, aber eine Art von Unterstützung kann und muß das System bieten und Benutzer müssen diese Unterstützung auch als solche empfinden und sich nicht durch das System eingeschränkt fühlen.

2.4 Verteilte Künstliche Intelligenz

Die Künstliche Intelligenz ist traditionell auf einzelne Systeme konzentriert. Aus mehreren Gründen strebt die Künstliche Intelligenz zur Verteilung. Ganz profan ergibt sich dieser Wunsch aus dem Fortschritt in den verfügbaren Computersystemen, welcher aus der Vernetzung vieler, unabhängiger Einzelsysteme besteht. Die kurze Geschichte der Künstlichen Intelligenz zeigt, daß sich viele Probleme als sowohl räumlich wie auch zeitlich inhärent verteilt strukturieren. Ein Beispiel ist eine Fabriksteuerung mit einer Vielzahl von kooperierenden Robotern. Des weiteren haben sich in der Informatik allgemein die Vorteile der Modularisierung von Software durchgesetzt und die Annahme ist, daß ein verteiltes Künstliche Intelligenz-System modulares Design eher unterstützt. Es zählen aber auch epistemologische Gründe für die Arbeit mit verteilten Systemen: Kooperation und Koordination sind wenig verstandene Phänomene. Theorien über diese Begriffe können sich mit Hilfe von Computermodellen analysieren lassen. Insbesondere die menschliche Kooperation möchte die VKI durch den Computer unterstützen [vM92]. Als weiterer Grund tritt auch hier wieder der Begriff der Mensch-Maschine-Interaktion auf. Die VKI versucht mit eigenen Ansätzen hier Lösungen zu schaffen [vM92].

Die beiden Extreme in der VKI bilden das *verteilte Problemlösen* und die *Multiagenten-Systeme*. Das erstere ist eine Lösung für ein spezielles Problem, in dem die Interaktionsstrategien (Koordination, etc.) als integraler Bestandteil des Systems vorgegeben sind. Das letztere besteht aus autonomen Agenten und betrifft die Frage, wie diese autonomen Agenten koordiniert werden.

In der *Koordination* betrachtet die VKI die Aktionen von Agenten und die Interaktionen zwischen Agenten. Sie sollten so gewählt werden, daß Leistungseigenschaften, welche das gesamte System betreffen, verbessert werden.

Auch für die VKI sind *Verhandlungen und Kommunikation* wesentliche Themen. Agenten haben z.B. unterschiedliche Ziele, Pläne und Semantiken. Sie verhandeln zur Auflösung von Konflikten, zur Problemerlegung und zur Aufgaben- und Ressourcenzuweisung. Die VKI sieht die Agenten dabei als gleichberechtigt an. Als Kommunikationsverfahren unterscheidet die VKI bisher nur die Message-Passing-Verfahren und die Blackboard-Verfahren.

Technisch werden folgende Aspekte betrachtet. Die Modellierung der Agenten teilt sich auf in eine technische Perspektive, die Architektur, und eine philosophische Perspektive. In der Architektur existieren zwei Strömungen. Die *reflektiven Agenten* besitzen Wissen und explizite Darstellungen über ihre Umwelt. Sie interagieren mit ihrer Umwelt über explizite symbolische Sprachen. Die *reaktiven Agenten* kennen keine expliziten Darstellungen zur Abspeicherung von Informationen und reagieren lediglich auf Umgebungseinflüsse. Reaktive Agenten kommunizieren nicht explizit. Sie interagieren lediglich über Stimuli mit ihrer Umwelt. Agenten sollten über beide Fähigkeiten verfügen.

In der *theoretisch, philosophischen Agentenmodellierung* betrachtet man das Bild, welches Agenten von sich und ihrer Umwelt machen. Dies betrifft Wünsche, Ziele, Absichten und Präferenzen (*intentions* und Wissen und Annahmen (*knowledge* und *belief*).

Während das traditionelle Planen von *einem* Planer ausgeht, werden beim *verteilten Planen* mehrere Planer berücksichtigt. Dabei wird unterschieden zwischen zentralisiertem Planen, wo ein Agent die Arbeit anderer Agenten plant, und dezentralem Planen, bei dem sowohl der Planungs- als auch der Ausführungsprozeß verteilt sind.

Neben den genannten Aspekten werden natürlich auch *Werkzeuge und Testumgebungen* untersucht. In dieses Teilgebiet fallen Sprachen zur Programmierung parallel aktiver Agenten, Sprachen, welche die Parallelität der Agenten auf Betriebssystemebene unterstützen, Werkzeuge zur Modellierung von Agenten, Kontrollstrukturen und verteilten Wissensbasen, Werkzeuge zur Entwicklung und Verifikation von Protokollen, Testumgebungen zur Entwicklung und Simulation von VKI-Systemen, und Shells, die ein bestimmtes Paradigma einer Systemarchitektur (z.B. Blackboards) unterstützen.

Signaling in ATM Networks — Features and Protocols

Burkhard Stiller⁹

1 Introduction

The general picture for the interoperation of communication protocols in an emerging ATM (Asynchronous Transfer Mode) environment providing user-to-user communications is stated in the B-ISDN (Broadband-Integrated Services Digital Network) Protocol Reference Model. *Signaling* — being part of the ATM control and management plane as well — is the critical issue of interoperability in ATM networks, *e.g.*, between digital voice devices, IP (Internet Protocol) hosts, or LAN (Local Area Networks) interworking devices. Signaling is necessary for finally transferring user data between ATM end-systems within well defined ATM connections while dynamically generating outgoing call set-ups and clearings, accepting incoming call set-ups and clearings, processing status maintenance and notification tasks, and running connection control issues according to user requirements for modern services. The underlying ATM concept of virtual path identifiers and virtual channel identifiers is “materialized” into a set of supporting signaling procedures and messages (exchanged between ATM switches and ATM end-systems as well) that are summarized as *call and connection control* issues.

A network scenario as presented in Figure 37 is quite common, where private (corporate) and public ATM switches are interconnected. Two different types of *User-Network Interfaces (UNI)* are distinguished. Private UNIs are located between ATM end-systems and private ATM switches. Instead, public UNIs are located between ATM end-systems and public ATM switches. Public ATM switches may define a public ATM network, where private ATM switches define a private ATM network. A main difference involves the administrative responsibility for all domains. While private domains may follow a local management scheme, which offers only dedicated interconnection points to the outside world, public domains have to offer well-defined interconnection points and signaling systems as well. Additionally, the *Network-Node Interface (NNI)* between private or public ATM switches is not visible to any ATM end-system or user.

A wide range of signaling protocols for ATM-based networks has been developed, in each case leading to the special focus of various working bodies (such as ITU-T or ATM-Forum) or vendors, while most of the protocols are still in the progress of definition or standardization.

2 Key Issues in Signaling for ATM-based B-ISDN

Relevant protocols for ATM are defined within a generic protocol architecture, the *B-ISDN Protocol Reference Model* [ITU93h], which contains orthogonal dimensions for different layers and various functions. Horizontal layers encompass:

- *Physical Layer*, specifying media technology-dependent issues;
- *ATM Layer*, including ATM specific functions, such as the generation of ATM cell headers, the (de-)multiplexing of cells, and mapping functions for identifiers;
- *ATM Adaptation Layer (AAL)*, including AAL specific functions, such as segmentation/reassembling of higher layer protocol data units into ATM cells, and convergence functions; and
- *Higher Layers* for application specific functions.

The vertical structure defines the *User Plane*, the *Control Plane*, and the *Management Plane*. User and control plane use identical physical and ATM layers respectively, but use different AAL and higher layers. Especially the user plane is responsible for transmission of user data, where the control plane handles all relevant issues on *Signaling*. This is the set-up, maintenance, and clear of calls and connections, while supporting different kinds of unicast, multicast, broadcast, and multipeer communication scenarios, negotiation and renegotiation of QoS (Quality-of-Service) parameters during set-up, admission control functions, ongoing QoS monitoring during the data transfer phase, and routing of set-up requests through the network. Finally, the management plane encompasses relevant functions to interact and coordinate between user and control plane activities.

⁹ The author is currently on leave from Universität Karlsruhe, Institut für Telematik, D-76128 Karlsruhe, Germany, and is sponsored by the Commission of the European Communities as a Research Fellow under the Human Capital and Mobility Scheme (RG 19372) at the University of Cambridge, Computer Laboratory, Systems Research Group, New Museums Site, Pembroke Street, Cambridge CB2 3QR, England, U.K. This present report is a brief summary of “A Survey of UNI Signaling Systems and Protocols for ATM Networks”, which is currently under construction.

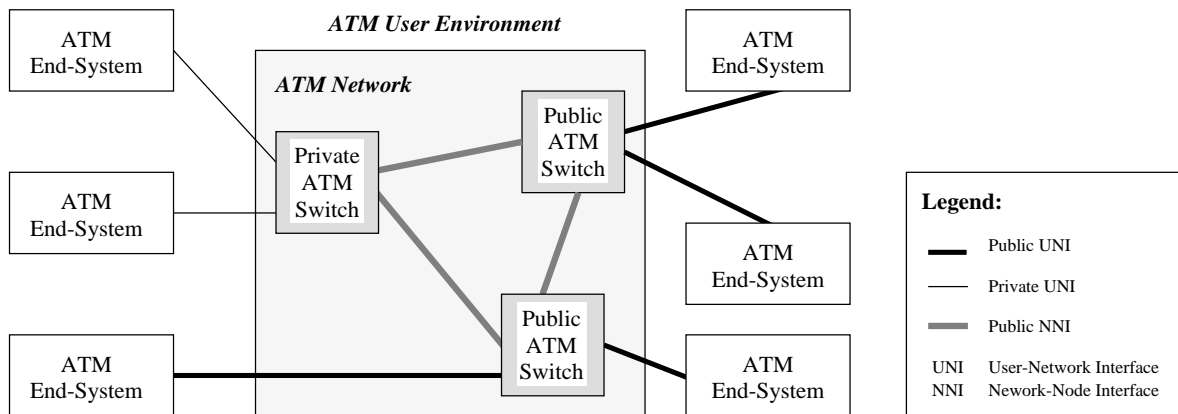


Abbildung 37. A Common Scenario for Interconnected ATM Switches and ATM End-systems

2.1 Signaling Protocol Architecture

A necessary prerequisite for transferring signaling messages is a reliable network service. Therefore, a SAAL (Signaling ATM Adaptation Layer) based on AAL type 5 [ITU94j] has been defined in the Recommendations Q.2100 [ITU94b], Q.2110 [ITU94c], Q.2120 [ITU93a], and Q.2130 [ITU94d]. *E.g.*, Q.2931 [ITU94e] runs on top of SAAL, which in turn runs on top of the ATM layer. Different types of AAL may be used as well, *e.g.*, SPANS (Simple Protocol for ATM Network Signaling) [For92] runs on top of either AAL type 3 or type 5. Moreover, the UNI signaling may be processed on top of a TCP/IP (Transmission Control Protocol/Internet Protocol) [DAR81b], [DAR81a] stack and may be used for transmitting signaling messages, to ensure the reliable transfer of these messages, as it is done by the Q.93B-based Netcomm signaling protocol [Net93]. Finally, signaling protocols may not require reliable signaling message transfer services at all, when the application itself is responsible for dealing with errors, such as corrupted, lost, or duplicated signaling messages.

For NNI signaling in B-ISDN a specific B-ISUP (Broadband-ISDN User Part) ([ITU93b], [ITU93c], [ITU93d], [ITU93e], [ITU93f], and [ITU93g]) based on the ISUP (ISDN User Part) ([ITU94n], [ITU94o], [ITU94p], and [ITU94q]) of the *Signaling System No. 7 (SSN 7)* — defined completely in Q.700-Q.782 [ITU94l] — is located on top of the MTP 1, 2, or 3 (Message Transfer Part) [ITU94m] of SSN 7. This regular layer 3 of SSN 7 runs on top of the already mentioned SAAL. Finally, meta signaling is defined in Q.142x [ITU94a] and done directly inside the ATM layer, as an internal extension of the ATM layer.

2.2 Layering and Signaling Tasks

Any signaling system has to hide completely any internals of the network and its topology from the user perspective. That is an important reason for the definition of the User-Network Interface, where the user only recognizes access methods that are handled with the “first” connected ATM switch, but has no knowledge on, *e.g.*, how to reach the potential receiver somewhere connected to the network. In general, signaling in ATM has to be regarded within three different layers:

- *Layer 1 Signaling* is done between physical hardware devices;
- *Layer 2 Signaling* allows for the interconnection of Interworking Units, *e.g.*, between LANs, using layer 2 endpoints; and
- *Layer 3 Signaling* supports the establishment of calls and connections between digital voice devices, IP or IPX hosts, relying on layer 3 endpoints.

Therefore, UNI signaling in the B-ISDN environment is located in layer 3 and defines peer functions for the network connection control. Furthermore, signaling issues may be separated into two distinct control areas:

- *Connection (Bearer) Control* defines procedures to set-up or initialize features of the user data connection, *e.g.*, the ATM connection, or the process of connecting that type of connection.

- *Call Control* defines procedures for maintaining the connection itself, *e.g.*, associating specific Virtual Path Identifiers (VPI) or Virtual Connection Identifiers (VCI) with a calling user, calling of any destination, or clearing VPI/VCI tables.

Two important tasks exist in any signaling scenario: network-dependent and service-dependent signaling tasks. The specific target of signaling within ATM networks includes two main ATM *network-dependent tasks*, only:

- Set-up, maintenance, and clear of Virtual Channel and Virtual Path Connections and
- negotiation of traffic characteristics.

Service-dependent tasks are in general completely independent of any specific network feature and are relevant for signaling systems. But they are *not* compulsory for signaling systems and *may* only be integrated in the signaling system definition approach, *e.g.*:

- The definition of multicast and multipoint scenarios and its direct support;
- the symmetric or asymmetric behavior of connections; or
- QoS parameter negotiation (such as bandwidth, latency, or error rates) for services.

Unfortunately, in case of ATM, QoS is related to services *and* the network as well, *e.g.*, any service requirement regarding throughput has to be mapped onto the network specific ATM peak cell rate, sustainable cell rate, or maximum burst size. Therefore, a complete signaling system *shall* include network-dependent tasks *and* service-dependent tasks as well.

2.3 Virtual Signaling Channels

In ATM-based B-ISDN networks *Out-of-Band Virtual Signaling Channels (VSC)* are used. They form the main basis for processing signaling tasks, *e.g.*, transmitting and receiving signaling control messages. In general, three different types of VSCs are distinguished:

- Meta Virtual Signaling Channel (MVSC);
- Broadcast Virtual Signaling Channel (BVSC); and
- Point-to-Point Virtual Signaling Channel (PVSC).

MVSCs are bidirectional and are used to establish BVSCs or PVSCs as necessary. These are used to signal all types of ATM connection signaling messages between different ATM end-systems and ATM switches. BVSCs use to be unidirectional, while PVSCs are bidirectional. At least, one VSC of either type is mandatory and, therefore, permanent for each single UNI. However, one static VSC, regardless of which type, has to be assigned at a defined (pair of) VPIs and VCIs initially for each ATM end-system.

3 Existing Signaling Systems and Protocols

Existing signaling systems and protocols are listed in an overview in Table 5, which is not complete, but presents several relevant approaches.

The International Telecommunications Union – Telecommunications Standardization Sector (ITU-T) Recommendations for the Signaling System No. 7 (SSN 7) in the Q.700-series [ITU94l] encompass the N-ISDN (Narrowband-Integrated Services Digital Network) version of an NNI signaling system. For N-ISDN the ITU-T Recommendation Q.930 [ITU94r], Q.931 (similar to I.451) [ITU94s], and Q.93B [ITU92] has been introduced as an UNI signaling system. Nevertheless, the *Digital Subscriber Signaling System No. 1 (DSS 1)* could not be applied to ATM-based B-ISDN networks directly, since it did not include schemes for dealing with virtual paths and virtual channels. Therefore, the *Digital Subscriber Signaling System No. 2 (DSS 2)* in the Recommendation Q.2931 [ITU94e] (as well as extensions, such as multicast [ITU94i] and Quality-of-Service parameter negotiation [ITU94f], [ITU94g], [ITU94h]) has received final preparation. ETSI (European Telecommunications Standards Institute) is in progress of defining a UNI signaling protocol, which is subdivided into *basic call* [ETS92a] and *supervisory call* [ETS92b] issues.

The ATM-Forum defined the UNI Specification 3.0 [ATM93], now being updated in UNI Specification 3.1 [ATM94a], which is not interoperable with the former version. These specifications are based on subsets and extensions of the above mentioned ITU-T Broadband Access Signaling Protocol Recommendations. A proposal for UNI 4.0 is under discussion [ATM94b]. Additionally, examples of vendor specific signaling protocols are Fore's interim signaling protocol SPANS [For92], [For93], a Q.93B-based

Organization	Standard	Content	Remarks
ITU-T	Q.701-Q.707	Message Transfer Part	SSN 7
	Q.711-Q.716	Signaling Connection Control Part	SSN 7
	Q.761-Q.766	ISDN User Part	SSN 7
	Q.771-Q.775	Transaction Capabilities	SSN 7
ITU-T	Q.930/Q.931	N-ISDN Voice Standard	DSS 1
ITU-T	Q.93B	Basic Call	Preliminary DSS 2
ITU-T	Q.2931	Point-to-Point	DSS 2
	Q.298x	Point-to-Multipoint	DSS 2
	Q.29xy	Various additional features	DSS 2
ETSI	DE/SPS-5024	Basic Call	
	DE/SPS-5034	Supervisory Call	
ATM Forum	UNI Spec 3.0	Point-to-Point and Point-to-Multipoint	
	UNI Spec 3.1	Point-to-Point and Point-to-Multipoint	
Fore Systems	SPANS	Point-to-Multipoint	For ATM LANs
Netcomm	Q.93B-based	Point-to-Point	
NetExpress	FAST Select	Fast Call Set-up procedure	
Bellcore	EXPANSE	Multimedia Services	
AT&T Bell Laboratories	GSP	Connection Management	For various environments
Washington University	CMAP	Multipoint-to-Multipoint	
University of Melbourne	gNET	Point-to-Multipoint	For ATM LANs
University of Cambridge	MSNL-CM	RPC-based Message Transfer	For Fairisle

Tabelle 5. Overview of Several Signaling System Drafts, (Pre-)Standards, and Definitions

protocol from Netcomm [Net93], and FAST Select [RMDB92]. Furthermore, Bellcore worked on EXPANSE in support of complex multimedia services [Min91], and Bell invented GSP (Generic Signaling Protocol) [MT92a], [MT92b]. Finally, university work has been driven by project requirements, such as with CMAP (Connection Access Management Protocol) [BDG91], [CD94], [DGB92], gNET [Arm94], and MSNL-CM (MultiService Network Layer-Connection Management) [Cro93], [Cro94].

3.1 Signaling Capabilities and Messages

User communications in an ATM network takes place after a set-up of an ATM connection, which is based on signaling messages transmitting set-up request information to one or multiple destinations. Depending on the request, either the network (an ATM switch) or the destination (an ATM end-system) agrees upon the information included in the message. A set-up includes other tasks to be managed, *e.g.*, the selection of an appropriate Virtual Path, the mapping of Virtual Channel Identifiers and Virtual Path Identifiers in intermediate switches, resource negotiations between switches, and the call and connection control in the calling and called ATM end-system.

Signaling protocols, such as Q.2931/Q.298x, UNI 3.1, SPANS, and CMAP, include the specification of procedures and messages to achieve *capabilities*, which define the purpose of signaling in various scenarios. They encompass, *e.g.*, switched connections for point-to-point as well as point-to-multipoint connections that support traffic having symmetrical or asymmetrical bandwidth requirements.

A *message* acts as a pre-defined function for exchanging signaling information, that is further refined into information elements. Examples are listed in Table 6 defining messages of two classes and six types within Q.2931/Q.298x, UNI 3.0/3.1, and SPANS, while Table 7 contains CMAP messages. The structure of each message, possibly consisting out of several variable length information elements, contains besides others one important information for the ITU-T and ATM-Forum case: The *Call Reference* consists out of a length field, a call reference flag, and the value itself. It is used — instead of an end-to-end significance — as a local identification of the call. The call reference value is fixed for the lifetime of a call at the originating side. Since multiple calls may be processed within one virtual signaling channel, the call reference value distinguishes messages from different calls. The flag defines, whether the message is being sent from the side that originated the call reference or from the responder.

Class	Type	Q.2931/Q.298x	UNI 3.0/3.1	SPANS
Point-to-Point	Set-up	Alerting	—	—
		Call proceeding	Call proceeding	Open request
		Connect	Connect	Open indication
		Connect acknowledge	Connect acknowledge	Open response
		Set-up	Set-up	Open confirm
	Clearing	Release	Release	Close request
		Release complete	Release complete	Close indication
		—	—	Close response
		—	—	Close confirmation
	Managing	Notify	—	Status request
		Status	Status	Status indication
		Status enquiry	Status enquiry	Status response
		—	Restart	—
		—	Restart acknowledge	—
Point-to-Multipoint	Set-up	—	—	Multi request
		—	—	Multi indication
		—	—	Multi response
		—	—	Multi confirmation
	Joining	Add-connection	Add party	Add request
		Connection-added	—	Add indication
		Connection-added-acknowledge	Add party acknowledge	Add response
		—	Add party reject	Add confirmation
	Leaving	Release-connection	Drop party	Rclose request
		Release-connection-complete	Drop party acknowledge	Rclose indication
		—	—	Rclose response
		—	—	Rclose confirmation

Tabelle 6. Messages for Q.2931/Q.298x, UNI3.0/3.1, and SPANS

Every single information element contains information that are necessary for the network and/or the responder, relating to the processed call. Certain coding rules for the information elements are defined. A general information element format includes identifiers for information elements, such as “Broadband-Sending Complete”, “ATM Traffic Descriptor”, or “Quality-of-Service Parameter”. *E.g.*, the “ATM Traffic Descriptor” contains the forward peak cell rate and the backward peak cell rate as defined in I.371 [ITU94k]. Reasons for a message and relevant diagnoses are transmitted by “Cause”. The “End-to-End Transit Delay” information element defines the selection of the end-to-end transit delay, while the “Connection Identifier” allows for the specification of the controlled connection. A “Number” defines a network addressing, whereas a “Subaddress” defines full addressing.

4 An Approach for a Global Signaling System Architecture

Since existing signaling systems and protocols do not offer a similar set of capabilities, messages, and information elements, the following list includes analyzed aspects for a generic signaling system comprising a broader capability- and services-based view of important signaling issues:

- Support of switched and permanent channel connections, including single and multiple connections per call in (multi-)point-to-(multi-)point communication scenarios.
- Definition of basic signaling functions via similar signaling protocol messages, information elements, and procedures for each communication scenario.
- Support of uni/bidirectional and symmetric/asymmetric connections concerning QoS requirements, such as bandwidth or error rates for each communication scenario.
- Provision of information for resource negotiation and reservation during connection/call set-up (admission control) and renegotiation during data transfer, including resource monitoring, such as for QoS parameters.

Class	Type	CMAF	
Multipoint-to-Multipoint	Set-up	open_call verify_mod_ep	invite_add_ep announce_add_ep
	Clearing	close_call	announce_close_call
	Managing	mod_call	announce_mod_call
		mod_con	announce_mod_con
		mod_ep	invite_mod_ep
		verify_mod_ep	announce_mod_ep
		trace_call	trace_ep
		change_owner	invite_change_owner
		announce_change_owner	change_root
	Joining	announce_change_root	status
		alert	client_reset
		network_reset	error_report
		add_con	invite_add_con
		verify_mod_ep	announce_mod_con
	Leaving	add_ep	invite_add_ep
		verify_add_ep	announce_add_ep
		drop_con	announce_drop_con
		drop_ep	announce_drop_ep

Tabelle 7. CMAF Messages

- Using a highly dynamic call model, including detailed characterizations of calls, such as accessabilities or identifications of originator.
- Multiple simultaneous call and connection set-ups for each communication scenario.
- Support of simultaneous joining/leaving of one or multiple calls/connections into/from existing ones (user-driven and network-driven), while supporting mandatory and optional network connections.
- Using a final call and connection clearing.
- Support of all ATM transport services as well as other data link services.
- One statically defined virtual meta signaling channel, providing the facility to dynamically establish broadcast- and/or point-to-point-virtual signaling channels, or associating a pre-defined static point-to-point virtual signaling channel.
- Support of public and private UNI addressing formats for unique identification of ATM endpoints (such as E.164, OSI-NSAP, IEEE 802.6, or IP addresses).
- Definition of client registration mechanisms/ services for exchange of addressing information across UNIs.
- Specification of error detection and recovery (such as alerting and resetting).
- Provision of end-to-end parameter compatibility and identification, *e.g.*, for QoS parameters or signaling parameters, including target platform independent (such as from the operating system or the equipment used) encoding rules for all parameters.
- Independent specification of the signaling system and protocol of the underlying reliable network service (such as with AAL type 3, AAL type 5, SAAL or TCP/IP).
- Independence of the requested service (*e.g.*, multimedia, client/server, or RPC) and of the used protocol for transmitting user data, while supporting different subscriber equipment.

This catalogue should be taken into account for a future development of a *Global Signaling System Architecture*, which is based on distributed system architectures. It includes general mapping schemes and rules for different signaling message types and message contents; on one hand for the integration and experimentation of many different signaling systems and protocols, and on the other hand for signaling simplifications of complex communication and service scenarios. As the variety of existing signaling systems and protocols does not seem to converge to a single signaling system or protocol, an overlaid global signaling system should allow for an interoperable, different ATM networks embracing approach.

Acknowledgements: Thanks go to Kobus van der Merwe, who read earlier drafts of the “Signaling Survey” and commented on several aspects to help improving this final version.

Visualisierung vernetzter Systeme

Volker Vogelmann

1 Einleitung

Im Rahmen der Simulation und Visualisierung vernetzter Systeme lassen sich zum einen die Untersuchung der Scalability durch Bestimmung der Anzahl der Backbones¹⁰ bei wachsender Anzahl der Teilnehmer, die Identifizierung der Standorte durch Visualisierung des Netzes und die Bestimmung der Hierarchie der Routen und der alternativen Routen als *Optimierungsprobleme* identifizieren. Zum anderen stellt das Layout eines Graphen¹¹ eine *Optimierung* unter ästhetischen Gesichtspunkten dar. Entwurfswerkzeuge, welche diese Optimierungen unterstützen, erleichtern die Prozesse der Planung, Modellierung, Simulation und Analyse der Netze erheblich. Graphische Entwurfswerkzeuge und Benutzerschnittstellen erlauben immer häufiger die direkte Manipulation graphischer Objekte. Komplexe Sachverhalte werden oft durch Graphen visualisiert. Anforderungen an diese Werkzeuge sind das Lösen von np-vollständigen und Optimierungsproblemen aus der Vernetzungs- und Graphentheorie, insbesondere automatisches Layout. Lösungsstrategien der kombinatorischen Optimierung sind u.a. evolutionäre Algorithmen, Simulated Annealing und Simulated Annealing.

2 Kombinatorische Optimierungsprobleme und ihre algorithmische Lösung

Jedes Problem der kombinatorischen Optimierung stellt ein Suchproblem in einem komplexen Suchraum dar, wobei jedem Punkt im Suchraum vermöge einer Funktion ein Zahlenwert zugewiesen wird. Dabei liegt der Suchraum meist als Menge diskreter Punkte (z.B. n-Tupel natürlicher Zahlen) und nicht in Form stetiger Intervalle vor, sodaß Differentiale und Stetigkeit im Sinne der Analysis keine anwendbaren Begriffe darstellen. Um Optima (Extrema) zu finden, eignet sich daher die Schulmethode nicht, bei der man die Ableitungsfunktion Null-setzt. Zum Auffinden globaler oder zumindest fastglobaler Optima wurden zahlreiche algorithmische Verfahren entwickelt. Obwohl diese nach außen hin sehr unterschiedlich wirken, arbeiten die meisten von ihnen nach den gleichen fundamentalen Prinzipien.

2.1 Deterministische Algorithmen

Deterministische Optimierungsalgorithmen verwenden in ihrem Ablauf keine Zufallselemente, sondern benutzen eine Heuristik, um die Suche zu steuern. Bei den meisten deterministischen Algorithmen ist die Suchheuristik nur auf ein spezielles, konkretes Optimierungsproblem anwendbar, während einige der neuesten deterministischen Algorithmen auf beliebige Probleme anwendbar sind. Die Heuristik, die sie verwenden, stellt kein Lösungsrezept für ein spezielles Optimierungsproblem dar, sondern es handelt sich um heuristische Verfahren, welche deterministische Regeln für die Suche im Lösungsraum angeben.

Threshold Accepting und der Simulated Annealing Sowohl beim TA als auch beim Simulated Annealing [DSW93] handelt es sich um zwei neuere deterministische Optimierungsalgorithmen, die sich auf beliebige Optimierungsprobleme anwenden lassen und die auch für große praktische Probleme sehr schnell sehr gute Lösungen liefern. Eine informelle Darstellung des TA sieht so aus :

1. Wähle in der gegebenen Aufgabenstellung irgendeine Lösung (einen Zustand x im Zustandsraum X) als Ausgangspunkt.
2. Wähle einen Nachbarzustand y von x , der durch eine lokale Veränderung aus x hervorgeht.
3. Vergleiche die Zielfunktionswerte $f(x)$ und $f(y)$. Ist $f(y)$ um mehr als T (Threshold) schlechter als $f(x)$, so verwirf y ; bleibe bei x stehen und suche einen neuen Nachbarzustand von x . Ist aber $f(y)$ mindestens so gut wie $f(x) - T$, ist also $f(y)$ höchstens um T

¹⁰ Als Beispiel wird hier das Telekommunikationsnetz BERKOM betrachtet.

¹¹ Das aus der Menge der Knoten und der Menge der Kanten bestehende Paar, wobei eindeutig festgelegt ist, welche Knoten durch Kanten verbunden sind, wird *Graph* genannt. Ein Telekommunikationsnetz ist ein Beispiel für einen Graphen [Neu75].

schlechter als $f(x)$, so gehe zum Zustand y ueber. Wir sagen, y wird akzeptiert. Von y aus fahre weiter fort.

4. Wenn laengere Zeit keine Verbesserungen auftreten, senke T langsam auf null.

Beim *Sintflutalgorithmus* erlaubt man dem suchenden Wanderer im Zielfunktionsgebirge im Gegensatz zum TA eine Richtung auch dann einzuschlagen, wenn sie auch eine noch so große Verschlechterung mit sich bringt. Gleichzeitig regnet es beim Sintflutalgorithmus unaufhörlich auf das Zielfunktionsgebirge, so daß der Wasserstand langsam aber unaufhörlich steigt. Der Wanderer darf dabei nicht in's Wasser treten. Irgendwann bilden sich beim Sintflutalgorithmus immer mehr Inseln im Gebirge.

2.2 Stochastische Algorithmen

Im Gegensatz zu deterministischen Algorithmen, erlauben stochastische Algorithmen zufällige Zustandsübergänge im Zustandsraum. Solange dabei prinzipiell jeder Zustandsübergang mit einer, wenn auch noch so kleinen Wahrscheinlichkeit erlaubt ist, läßt sich bei diesen Algorithmen sogar beweisen, daß sie mit zunehmender Zeit gegen das globale Optimum streben. Diese Aussage hat aber nur akademischen Wert, denn das Erreichen dieses globalen Optimums kann bereits im Mittel unvertretbar viel Zeit in Anspruch nehmen. Für praktische Anwendungen ist vielmehr die Frage von Interesse, wieviel Zeit benötigt wird, um eine fast optimale oder eine akzeptable Lösung zu finden. Der Einsatz stochastischer Optimierungsalgorithmen ist daher besonders dann sinnvoll, wenn sehr hochwertige Lösungen gefordert werden oder wenn eine auf das Problem passende Heuristik nicht bekannt ist. Die wichtigsten stochastischen Optimierungsverfahren sind:

- neuronale Netze (Boltzmannmaschine)
- simulated annealing (simuliertes Kühlen)
- genetische Algorithmen

Die *Boltzmann-Maschine* ist eine Weiterentwicklung der Hopfieldnetze, die die simulated annealing Technik benutzt, um aus lokalen Minima wieder entkommen zu können. Eine Boltzmann-Maschine besitzt n Neuronen, von denen jedes mit jedem anderen über ein Gewicht verbunden ist. Im Gegensatz zu Hopfieldnetzen, wo jedes Neuron gleichzeitig als Eingabe- und als Ausgabeunit fungiert, wobei die Eingabe zum Zeitpunkt $t+1$ gerade die Ausgabe zum Zeitpunkt t des selben Neurons übernimmt, sind bei Boltzmannmaschinen besondere Eingabe- und Ausgabeneuronen vorhanden. Jedes Neuron produziert als Ausgabe entweder '0' oder '1'.

Beim *simulated annealing* handelt es sich weder um einen konkreten Algorithmus, noch um eine Entwurfsmethode, sondern um einen *Metaalgorithmus*. Das zu optimierende System wird bei einer hohen 'Temperatur' zum Schmelzen gebracht und anschließend wird die Temperatur in kleinen Schritten gesenkt. In jedem Iterationsschritt wird das System in Hinsicht auf eine gegebene Kostenfunktion optimiert. Der Algorithmus endet mit dem Erreichen eines vorgegebenen Wertes der Kostenfunktion oder bei einer vorgegebenen maximalen Rechenzeit. *Genetische Algorithmen* (GAs) orientieren sich an der Art und Weise, wie in der Natur die Evolution von Pflanzen oder Lebewesen vonstatten geht. Die Berechtigung die Evolution als Lösungsvorlage von Computerproblemen zu benutzen, ergibt sich aus vielzähligen Beispielen von Lebewesen und Pflanzen, welche sich im Laufe der Zeit immer wieder optimal an ihre Umgebung und Umwelteinflüsse angepaßt haben [Sch94b]. Die grundlegende Idee eines genetischen Algorithmus ist eine bestimmte Anzahl von Lösungsansätzen zu verwalten und diese dann in dem Maße zu verändern, daß einer oder mehrere „hoffentlich“ gegen die korrekte Lösung streben [Gol89]. Ein großer Vorteil von genetischen Algorithmen ist, daß sie sich sehr leicht und effizient verteilen lassen [RB92].

2.3 Gemeinsamkeiten und Unterschiede von Optimierungsverfahren

Steuerfunktion Die Begriffe Fitnessfunktion, Zielfunktion, Energiefunktion und Kostenfunktion umschreiben eigentlich ein und dasselbe: Funktionen über einem Zustandsraum, von denen wir ein globales Optimum suchen.

Hillclimbing Algorithmen Könnte man den Graph einer Steuerfunktion über dem n -dimensionalen Zustandsraum aufmalen, so bekäme man ein entsprechendes Funktionsgebirge mit Bergen, Tälern, lokalen Extrema, Sattelpunkten usw. Sowohl TA als auch der Sintflutalgorithmus und alle hier vorgestellten stochastischen Algorithmen gehören zu den sogenannten Hillclimbing Algorithmen, da man sich die

Abbildung 38. Graph G1

Lösungssuche so vorstellen kann, daß ein oder mehrere Wanderer im Funktionsgebirge Berge ersteigen und hoffen, irgendwann den höchsten Berg (das globale Maximum) zu finden. Eine interessante Frage ist:

Welche Eigenschaften können solche Gebirge haben, welche können ‘Wanderer’ in die Irre führen, und wie können sich ‘Wanderer’ in unübersichtlichen Gebirgen orientieren?

Flucht aus lokalen Optima und Vermeidung des mehrfachen Betretens desselben lokalen Optimums Sämtliche stochastischen Algorithmen verwenden Zufallsentscheidungen, um aus lokalen Optima zu entkommen. TA und der Sintflutalgorithmus benutzen eine Heuristik, um aus lokalen Optima wieder herauszufinden. Es gibt einen Punkt, der jedoch bis jetzt die stochastischen von den heuristischen Algorithmen in ihrer Fähigkeit, lokale Optima verlassen zu können, weitgehend unterscheidet. Deterministische Algorithmen können lokale Optima nur dann verlassen, wenn die Optima nicht zu hoch, bzw. zu tief sind oder wenn sie nicht zu steile Flanken besitzen. Stochastische Algorithmen finden aus jedem lokalen Optimum wieder heraus.

Temperatur Die Idee des Auskühlens besteht darin, zu Beginn noch große, sprunghafte Positionswechsel im Zustandsraum zu erlauben, während mit zunehmender Zeit nur noch immer kleiner werdende Wechsel bzw. große Wechsel mit immer weiter abnehmender Wahrscheinlichkeit zugelassen werden. Wenn man die anderen Algorithmen genau betrachtet, so fällt einem auf, daß beinahe jeder von ihnen ebenfalls einen der Temperatur vergleichbaren Parameter entweder explizit oder implizit mit sich führt. Implizit bedeutet in diesem Zusammenhang, daß ein Parameter, der im Algorithmus eigentlich einem anderen Zweck dient, gleichzeitig als eine Art Temperatur dient, und der Wert dieses Parameters im Laufe der Zeit abnimmt.

2.4 Evolutionäre Algorithmen

Da es hauptsächlich für das Constrainthandling bei komplexeren Lösungsräumen sehr nützlich wäre, eine problemangepaßte Datenstruktur statt unstrukturierte Strings zu verwenden, bei der die Annahme ungültiger Zustände bereits durch die Datenstruktur selbst weitgehend ausgeschlossen ist, lag es nahe, genetische Algorithmen zu entwerfen, die auf derartigen Datenstrukturen operieren. Diese Modifikation genetischer Algorithmen nennt man *evolutionäre Algorithmen*. Abb. 38 zeigt z.B. das Layout eines Graphen, also einen Phänotyp [Mic92]. Abb. 39 zeigt die dazugehörige Genotypcodierung in einer quadratischen Zellrasterdatenstruktur, wie sie in einem evolutionären Algorithmus Verwendung fände. Bei der in Abb. 39 dargestellten Genotypcodierung kommen als Mutationsoperatoren das Vertauschen von Zeilen, das Vertauschen von Spalten, das Vertauschen zweier Knoten, Reihenfolgeinversion von Zeilen bzw. Spalten und einige weitere in Betracht.

3 Methoden der Layouterstellung

Das automatische, qualitativ hochwertige Bildschirmlayout von umfangreichen Graphen ist ein Anwendungsgebiet neueren Datums, da es im Dialogtrieb erhebliche Anforderungen an die verfügbare

Abbildung 39. Genetische Darstellung von G1

Rechenleistung stellt. Sehr gut erforscht hingegen ist das Gebiet des automatischen circuitlayout, da VLSI-layout bei der Komplexität heutiger Bausteine ohne Rechnerunterstützung nicht mehr denkbar ist [KO90].

Ästhetische Kriterien für die Güte eines Layouts sind:

- Vorzugsrichtung der Kanten (z.B. *nach oben* vermeiden)
- Gleichmäßige Verteilung der Knoten auf der Plazierungsfläche
- Wenig Überschneidungen der Kanten
- Möglichst kurze Kanten

3.1 Methoden der Layouterstellung

1981 schlug *Sugiyama* einen Algorithmus zum Graphenlayout vor, der geeignet ist um Graphen, die Hierarchiebeziehungen beschreiben, übersichtlich darzustellen [Sug81]. Er setzt einen gerichteten Graphen voraus. Der Sugiyama-Algorithmus hat den Vorteil, daß er sehr schnell ist. Seine Nachteile überwiegen allerdings bei weitem. Das von ihm produzierte Layout ist normalerweise schon ab 20 Knoten sehr unübersichtlich, da sich gerne lange Kanten unter sehr kleinem Winkel schneiden. Oft finden sogar mehrere solcher Schnitte in einem sehr eingegengten Bereich statt. Sugiyama kommt i.a. nicht mit der kleinsten Anzahl nötiger Überkreuzungen aus und findet auch bei planaren Graphen nicht immer eine planare Darstellung. Der Sugiyama-Algorithmus läßt sich nur sehr schwer an andere Anforderungen anpassen.

3.2 Circuitlayout

Problemuntergliederung Wenn Schaltkreise automatisch layoutet werden sollen, so wird diese Aufgabe immer in mindestens zwei aufeinanderfolgenden Schritten gelöst, da das Gesamtproblem zu komplex ist, um auf einmal gelöst zu werden. Die beiden Hauptteilaufgaben, in die das Gesamtproblem immer zerfällt, sind das Placement und das Routing (die Verdrahtung). Sowohl das Placement als auch das Routing können dabei in weitere, sequentiell zu lösende Teilprobleme zerlegt werden.

Placementalgorithmen Die Plazierungsalgorithmen unterteilen sich in die Gruppe der heuristischen und der stochastischen Algorithmen. Praktisch allen Plazierungsalgorithmen ist jedoch gemeinsam, daß sie, obwohl eine sehr fein gerasterte Layoutfläche zur Verfügung steht, die Plazierung zwecks Komplexitätsreduktion zuerst auf einem viel gröberen Raster mit verhältnismäßig wenigen Rasterpunkten durchführen. Auch der Sugiyama-Algorithmus verfährt auf diese Weise. Das feine Positionieren überlassen sie meist Finetuning-Algorithmen, die meistens erst nach der Verdrahtung angewandt werden. Es existiert eine ganze Reihe sehr unterschiedlicher Algorithmen um das Problem des cellplacement zu lösen:

- Kräftegesteuerte Plazierung
- Plazierung durch Partitionierung
- Min - Cut - Verfahren
- Simulated Annealing
- Genetische Algorithmen
- Evolutionäre Algorithmen

Routingverfahren Nachdem die Zellen plaziert sind, findet die Verdrahtung statt. Eines der ältesten und einfachsten Routingverfahren ist der *Lee-Algorithmus*. Lee-Routing besitzt den Vorzug, immer einen Weg zu finden, sofern einer existiert.

Eine andere Routingmethode stellt der *Hightower Algorithmus* dar, der zu den Line Search Algorithmen zählt. Es gibt Weiterentwicklungen des Hightower-Verfahrens wie z.B. das Line-Expansion-Verfahren und es gibt auch Algorithmen, die die Vorteile des Lee-Routing mit den Vorteilen von Line Search Verfahren kombinieren. Um das Routingproblem zu lösen, wird die freie Fläche zunächst in Kanäle aufgeteilt. Diesen Schritt nennt man *Channel Assignment*. Es ist Aufgabe des dritten und letzten Routingteilschritts, des Channel Routing, die Netzanschlüsse der einen Kanalseite mit den zugehörigen Netzanschlüssen auf der anderen Kanalseite so zu Verdrahten, daß keine Überkreuzungen entstehen und gewisse andere Kriterien so gut wie möglich eingehalten werden.

1 Einleitung

Neuartige Anwendungen wie beispielsweise Multimedia und die steigende Leistungsfähigkeit der Rechnersysteme führen zu verstärkten Anforderungen an die Vernetzung der Rechner. Wo vor zehn Jahren ein Ethernet mit 10 Mbit/s noch vollkommen ausreichend war, ist es heute zumindest zeitweise stark überlastet. Aus diesem Grund wurde die Installation einer neuen, leistungsfähigeren Technologie am Institut für Telematik beschlossen.

In den leistungskritischen Bereichen wurde die Ethernet-Vernetzung durch eine FDDI-Vernetzung ersetzt. Für die FDDI-Vernetzung war die Installation neuer physikalischer Medien notwendig. Die Wahl fiel dabei auf Glasfaser, um auch für zukünftige Entwicklungen gerüstet zu sein. Der Abschnitt 2 beschreibt die neue Infrastruktur und die darauf realisierte FDDI-Vernetzung im Detail.

Für die Kopplung der FDDI-Netze untereinander und mit den Ethernet-Netzen sind zwei Komponenten wesentlich. Sie werden im Abschnitt 3 vorgestellt.

Die Glasfaser-Infrastruktur ist vorerst hauptsächlich für die Nutzung zur FDDI-Vernetzung innerhalb des Instituts vorgesehen. Später läßt sie sich aber genauso für ATM nutzen. Die momentanen Planungen über die ATM-Ausstattung im Institut und ihre Anbindung an die restliche Vernetzung sowie an öffentliche ATM-Netze wird im Abschnitt 4 genauer vorgestellt.

2 Neue Netzinfrastruktur am Institut für Telematik

Bei der Erneuerung der Infrastruktur des Instituts für Telematik wurde darauf geachtet, daß in den leistungskritischen Bereichen die alte Ethernet-Vernetzung durch FDDI-Technologie ersetzt wird. Ferner wurde das vorher einzige Ethernet-Segment (nur über Repeater verbunden) in mehrere Segmente zerlegt. Die Abbildung 40 zeigt die wesentliche Struktur der neuen Vernetzung am Institut.

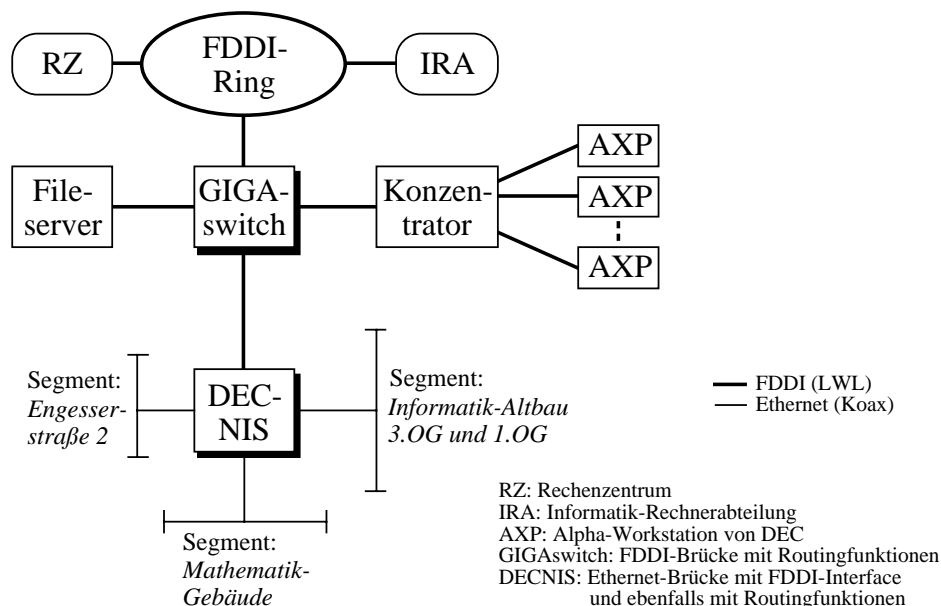


Abbildung 40. Die neue Netzinfrastruktur am Institut für Telematik

Die zentrale Komponente der neuen FDDI-Vernetzung bildet der GIGAswitch von DEC, der im Abschnitt 3.1 genauer vorgestellt wird. Der GIGAswitch ist mit einem Dual-Attachment-Port an den neuen FDDI-Ring der Informatik angeschlossen. Über diesen Anschluß hat das Institut für Telematik eine FDDI-Verbindung zur Informatik-Rechnerabteilung (IRA), zum Rechenzentrum (RZ) und somit

auch an die Außenwelt. An einem weiteren Port (Single Attachment Station, SAS) des GIGAswitch ist der Fileserver über eine Punkt-zu-Punkt-Verbindung angeschlossen. Auf diese Art steht beim Zugriff auf den Fileserver eine Bandbreite von ca. 140 Mbit/s zur Verfügung. Die leistungsfähigen Alpha-Workstations (AXP) werden über einen Konzentrator mit einem Port (SAS) des GIGAswitch verbunden. In dem Fall müssen sich die Workstations die Bandbreite von 100 Mbit/s teilen. Es ist aber auch möglich, einige Workstations direkt mit einem Port (SAS) des GIGAswitch zu verbinden, um ihnen so mehr Bandbreite zur Verfügung zu stellen. Schließlich wird ein Port (SAS) des GIGAswitch dazu benötigt, die Verbindung zum FDDI-Interface des DECNIS zu realisieren.

Der DECNIS ist für die Verbindung der neuen FDDI-Vernetzung mit der alten Ethernet-Vernetzung vorgesehen. Der Aufbau und die Funktionsweise dieser Komponente wird im Abschnitt 3.2 genauer beschrieben. Über sein FDDI-Interface ist der DECNIS mit dem GIGAswitch und somit mit der FDDI-Vernetzung verbunden. Neben dem FDDI-Interface stellt der DECNIS noch mehrere Ethernet-Ports zur Verfügung. Dort werden die Segmente der alten Ethernet-Vernetzung angeschlossen. Einen Port beansprucht dabei das Segment des Informatik-Altbaus (Geb. 20.20), welches dort das 3.OG und auch das 1.OG versorgt. Ein weiterer Port steht für das Ethernet-Segment der Engesserstraße 2 (E2) zur Verfügung und ein Port ist für den Anschluß des Ethernet-Segments vom Mathematik-Gebäude (Geb. 20.30) vorgesehen.

3 Zentrale Komponenten der neuen Netzinfrastruktur

Die neue Netzinfrastruktur enthält zwei zentrale Komponenten:

1. FDDI-Switch *GIGAswitch/FDDI*
2. Ethernet-Switch *DECNIS 600*

Die beiden Komponenten werden in den Abschnitten 3.1 bzw. 3.2 vorgestellt. Neben diesen Komponenten sind im Ethernet-Bereich noch einige Repeater zur Verlängerung der Segmente vorhanden.

3.1 Der GIGAswitch/FDDI

Der GIGAswitch/FDDI ist eine Hochleistungs-Netzwerkkomponente, die als Brücke mit integrierten Routingfunktionen charakterisiert werden kann ([SKO⁺94]). Er bietet bei vollem Ausbau Anschlüsse für insgesamt 34 FDDI-Ringe, die durch das Gerät bei Bedarf mit ihrer vollen Kapazität verbunden werden können. Dazu besitzt die Komponente eine leistungsfähige Switch-Architektur im Hintergrund, die eine Datenrate von maximal 3,6 Gbit/s bewältigen kann.

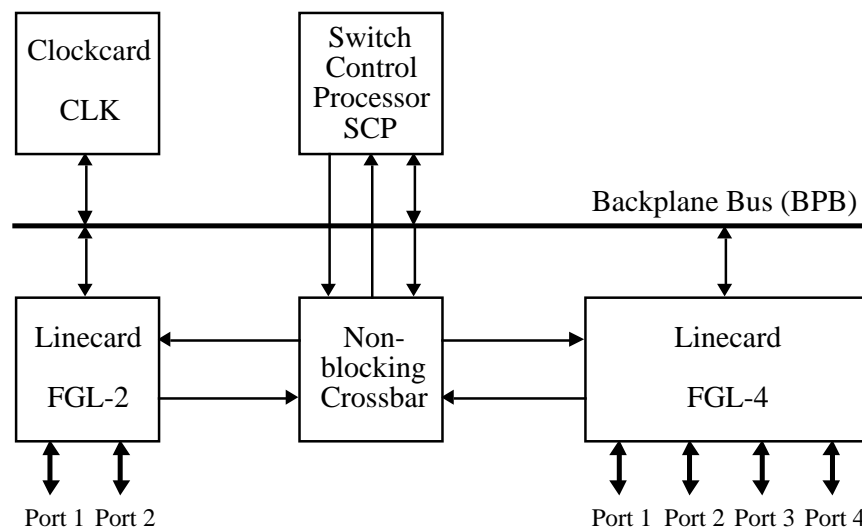


Abbildung 41. Die grobe Struktur des GIGAswitch/FDDI

Die Abbildung 41 zeigt die grobe Struktur des GIGAswitch/FDDI. Sie besteht aus einer Backplane, in die verschiedene Karten gesteckt werden können. Die drei notwendigen Karten sind der Non-blocking

Crossbar, der Switch Control Processor (SCP, [WO95]) und die Clockcard. Ferner können noch bis zu elf Linecards installiert werden, die je nach Typ entweder zwei FDDI-Ports (DAS oder SAS) oder vier FDDI-Ports (nur SAS) zur Verfügung stellen. An einem DAS-Port kann ein FDDI-Ring mit mehreren Komponenten angeschlossen werden, an SAS-Ports können hingegen einzelne Workstations mittels einer Punkt-zu-Punkt-Verbindung angeschlossen werden. Dabei kann über eine Punkt-zu-Punkt-Verbindung Full-Duplex FDDI gefahren werden, sofern die Workstation mit einem besonderen FDDI-Adapter ausgestattet ist. Es resultiert eine erreichbare Datenrate von 140-150 Mbit/s, da auf Token-Übertragung verzichtet wird und auf beiden Glasfasern permanent gesendet werden kann (für jede Richtung eine). Ferner bietet sich auch noch die Möglichkeit, einen Konzentrator an einen FDDI-Port (DAS oder SAS) des GIGAswitch/FDDI anzuschließen. Solch ein Konzentrator kann die Bandbreite des Ports auf mehrere, Punkt-zu-Punkt angeschlossene Workstations verteilen.

Primär trifft der GIGAswitch die Entscheidung über das Weiterleiten oder das Filtern von Dateneinheiten an Hand der Hardware-Adressen. Ist dem GIGAswitch aus früheren Dateneinheiten bekannt, wo sich die zur gesuchten Hardware-Adresse gehörende Station befindet, so wird die Dateneinheit nur an den entsprechenden Port weitergeleitet. Die dazu notwendigen Tabellen werden auf jeder Linecard unterhalten, so daß die Dateneinheit bei bekannter Hardware-Adresse direkt von der empfangenden Linecard über den Crossbar auf die sendende Linecard weitergeleitet werden kann (vgl. Abb. 41). Ist jedoch die Hardware-Adresse einer Dateneinheit auf der empfangenden Linecard nicht bekannt, so wird die Dateneinheit an den SCP weitergeleitet. Der SCP kann neben seiner großen Tabelle mit Hardware-Adressen auch noch auf gewisse Routing-Funktionen zurückgreifen, um eine Weiterleitungsentscheidung für eine Dateneinheit zu treffen. Hierzu schaut der SCP in die Nutzdaten der FDDI-Dateneinheiten hinein, um dort die IP-Zieladresse zu lesen. Auf der Basis der IP-Adresse kann der SCP IP-Routing durchführen, wobei er entweder an Hand seiner eigenen IP-Adressen und Subnetzmasken in Subnetze routet, oder aber seine statische Routingtabelle für IP-Adressen benutzt. Nur wenn der SCP auch keine Weiterleitungsentscheidung treffen kann, wird die betreffende Dateneinheit in alle angeschlossenen Netze geflutet. Der GIGAswitch läßt sich folglich als transparente Brücke mit Routingfunktionalität charakterisieren.

Neben dem Weiterleiten von Dateneinheiten bietet der GIGAswitch zahlreiche Möglichkeiten des Filterns. Im Normalfall werden vom GIGAswitch nur die Dateneinheiten gefiltert, von denen an Hand der Hardware-Adresse bekannt ist, daß das Zielsystem im gleichen Netzwerk liegt. Zur Realisierung von Sicherheitsmechanismen oder zur Reduzierung des Internetverkehrs können am GIGAswitch noch zahlreiche zusätzliche Filter eingestellt werden. Dabei kann sowohl nach Hardware-Adressen (Quelle und Ziel) als auch nach Protokollen (IP, Appletalk, etc.) gefiltert werden.

3.2 Der DECNIS 600

Der DECNIS 600 ist eine Ethernet-Brücke mit integrierter Routingfunktionalität ([BB93]). Die Komponente ähnelt dem GIGAswitch/FDDI in ihrem Aufbau, nur daß sie für Ethernet-Technologie ausgelegt ist. Sie besteht ebenfalls aus einem Backplane-Bus (Futurebus+), an den verschiedene Karten angeschlossen werden können. Zwei Karten, die Management Processor Card (MPC) und das Packet Random-Access Memory (PRAM), müssen vorhanden sein. Die restlichen sieben Steckplätze können mit Linecards bestückt werden. Die FDDI-Linecard für den DECNIS 600 beansprucht dabei zwei Steckplätze, Ethernet-Karten hingegen nur einen. Die FDDI-Linecard wird im Institut für Telematik dazu benutzt, über den DECNIS 600 die Verbindung zwischen der FDDI-Vernetzung und der Ethernet-Vernetzung herzustellen.

Die beiden Karten MPC und PRAM haben die folgenden Funktionen: MPC ist für das Management und die zentralen Funktionen des DECNIS 600, wie beispielsweise das Berechnen der Weiterleitungstabelle, verantwortlich. Die zentrale Speicherung aller Dateneinheiten im DECNIS wird vom PRAM übernommen. Ferner enthält das PRAM die zentrale Adreßerkennungsinstantz (ARE), welche die Weiterleitungsentscheidung an Hand der Zieladresse von Dateneinheiten trifft.

4 Zukünftige ATM-Ausstattung des Instituts für Telematik

Als zentrale Komponente der geplanten ATM-Ausstattung des Instituts für Telematik ist der GIGAswitch/ATM von DEC vorgesehen. Dieser Switch soll mit acht ATM-Ports ausgestattet werden, an die vorerst drei Alpha-Workstations mit Turbochannel-Bus und zwei SUNs (SPARCstation 10 und SPARCstation 20) angeschlossen werden. Ein ATM-Port des Switches wird für die Anbindung an das Rechenzentrum benötigt, um Verbindung zum geplanten BelWü-4 und zu den Telekom-Pilotnetzen zu haben. Die Abbildung 42 verdeutlicht die lokale Situation am Institut.

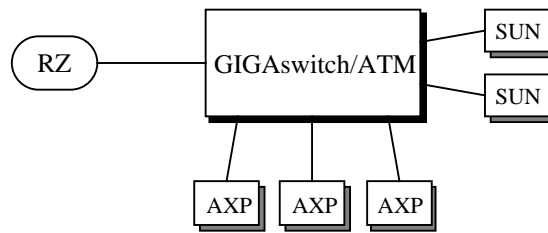


Abbildung 42. Die geplante ATM-Vernetzung

In der ersten Phase ist die beschriebene ATM-Vernetzung allerdings eine reine Testkonfiguration. Die klassischen Dienste wie beispielsweise Fileservices und Internet-Dienste werden weiterhin auf den betroffenen Rechnern über die FDDI- bzw. Ethernet-Vernetzung abgewickelt. Es sind allerdings zwei Versuche geplant, die ATM-Vernetzung in die restliche Vernetzung des Instituts zu integrieren. In einem ersten Schritt soll eine der Alpha-Workstations als Router aufgesetzt werden, um Dateneinheiten zwischen FDDI und ATM zu routen. Als zweite Möglichkeit ist eine ATM-Linecard für den GIGAswitch/FDDI zu sehen, über die der GIGAswitch/FDDI und der GIGAswitch/ATM auf Basis von ATM verbunden werden können. Allerdings bietet die momentan verfügbare ATM-Linecard für den GIGAswitch/FDDI noch nicht die erforderliche Funktionalität, um FDDI- und ATM-Netze auf gleicher Ebene zu koppeln. Zur Zeit realisiert diese ATM-Linecard nur die Kapselung von FDDI-Dateneinheiten über ATM-Netze. Es ist folglich nur ein Remote-Bridging von FDDI-Netzen über ATM-Netze möglich.

5 Zusammenfassung

In diesem Beitrag ist die neue, leistungsfähige Netzinfrastruktur des Instituts für Telematik vorgestellt worden. Als physikalisches Medium wurde Glasfaser ausgewählt, so daß auf diesem Medium sowohl FDDI als auch ATM realisiert werden können. Auch bei der Installation der Glasfaser wurde bereits auf zukünftige Entwicklungen Rücksicht genommen. Die Wahl einer sternförmigen Verkabelung für die momentane FDDI-Vernetzung läßt problemlos eine spätere Migration zur ATM-Technologie zu.

Leistungsfähige Netzwerkkomponenten, wie beispielsweise der GIGAswitch/FDDI, sind im Netzwerk integriert und realisieren eine Partitionierung des gesamten Netzwerks in Teilnetze. Auch die Anbindung des Institutsnetzes an die Außenwelt ist leistungsfähiger geworden. Die Anbindung über Ethernet an die IRA wurde durch eine FDDI-Verbindung zur IRA und zum Rechenzentrum ersetzt.

Mit der beschriebenen ATM-Ausstattung steht am Institut für Telematik in Kürze auch eine Testkonfiguration mit ATM-Technologie zur Verfügung. Dort können zum einen Erfahrungen mit der neuen ATM-Technologie gesammelt werden, zum anderen ist diese ATM-Ausstattung für Arbeiten innerhalb des Berkomp-Projektes notwendig.

Kommerzielle Nutzung von Netzen

Sonja Zwißler

Der Trend zur Kommerzialisierung des Internets und dessen Dienste nimmt immer konkretere Formen an. Prophezeiungen wie

*The employment and resource allocation changes generated by on-line shopping during the coming century might be comparable to the shifts out of agriculture during this century*¹²

werden dominanter. Mit der Kommerzialisierung rückt ein erhöhter Sicherheitsbedarf sowohl der Dienstanbieter als auch der Kunden in den Mittelpunkt des allgemeinen Interesses. Im folgenden wird ein System vorgestellt, das sich mit diesen Entwicklungen befaßt, und anhand dessen die Problemstellungen und die besonderen Anforderungen solcher Systeme aufgezeigt werden sollen. Es handelt sich hierbei um eine elektronische Ladenkette im Internet. Über diesen Dienst können sämtliche Produkte angeboten werden, die sich in elektronischer Form vertreiben lassen. Dieses Angebot ist für jedermann frei zugänglich. Der Vorteil für den Kunden liegt darin, daß er einen kompletten Einkaufsvorgang elektronisch abwickeln kann. Er kann sich über Produkte informieren, diese kaufen und erhält die bestellte Ware unverzüglich. Das hier vorgestellte System hebt sich durch seine Benutzerfreundlichkeit, Effizienz und vor allem durch seine Sicherheitsmechanismen von gegenwärtig existierenden Lösungen ab.

Im folgenden wird auf die Anforderungen und Entwurfsentscheidungen für die Realisierung dieser Idee eingegangen. Anforderungen bei der Entwicklung von Systemen, welche möglichst benutzerfreundlich, effizient und sicher sein sollen, liegen zunächst bei dem Verteilungsaspekt an sich. Konsistenz- sowie Verfügbarkeitsanforderungen stellen hohe Ansprüche vor allem an das Management der Daten. Diese Anforderungen fließen stark in das Systemdesign ein, da sie ein entscheidender Faktor zur Gewährleistung der Funktionalität des Systems bilden. Die Forderung nach Sicherheit bezieht sich nicht nur auf die verschlüsselte Übertragung vertraulicher Daten, sondern auch auf Authentisierung und Schutz vor Modifikation und Mißbrauch gespeicherter Daten. Die Aspekte der Systemsicherheit werden im folgenden gemäß der *IT-Sicherheitskriterien*¹³ untersucht. Sicherheit bedeutet im allgemeinen jedoch eine Erweiterung der Grundfunktionalitäten eines Systems um zusätzliche Aktionen, welche auch Zeit benötigen. Hier ist zu beachten, daß Effizienz eines Systems ein entscheidender Faktor für Benutzerfreundlichkeit und damit für dessen Akzeptanz bildet. Systemintern wirken sich die gehobenen Sicherheitsansprüche verstärkt auf die Administration des Systems aus. Der Wunsch auf sich ändernde Anforderungen flexibel reagieren zu können muß ebenfalls mit berücksichtigt werden.

Verteilung

Wird ein Vertriebssystem von zentraler Stelle organisiert, so kann es bei Übertragungen großer Produkte zu den Kunden zu großen Verzögerungen kommen. Der Extremfall ist die weltweite Auslieferung. Kommen öffentliche Netzwerke, wie das Internet zum Einsatz, so müssen große Lastschwankungen miteinbezogen werden. Durch unterschiedliche Zeitzonen kann für die Übertragung keine Auswahl weniger bevorzugter Zeiten getroffen werden. Werden solche Produkte immer wieder über große Distanzen übertragen, so führt dies zwangsläufig zu übermäßigen Belastungen der Netze. Um diese Nachteile zu verringern wird das System verteilt konzipiert, sodaß Kommunikation über größere Distanzen auf ein Minimum beschränkt wird. Ein weiterer Vorteil der Verteilung ist, daß durch den Einsatz mehrerer Filialen die Verfügbarkeit, die Robustheit und der Durchsatz des Gesamtsystems erhöht wird. Hierbei müssen strategisch interessante Knoten gewählt werden. Für deren Auswahl muß berücksichtigt werden, von welchen Knoten die häufigste Dienstbenutzung zu erwarten ist und wie sich dabei Engpässe im Netz möglichst umgehen lassen.

Durch die Wahl eines verteilten Ansatzes ergeben sich weitere Zielsetzungen. Zum einen sollen die Kunden in der Lage sein, den Dienst über beliebige Filialen in Anspruch zu nehmen. Andererseits müssen die einzelnen Knoten so gestaltet werden, daß die lokal Zuständigen selbst entscheiden, inwieweit sie Aufgabenbereiche übernehmen wollen oder nur als automatisierter Verteilungsstützpunkt auftreten wollen.

¹² Larry Press, "Commercialization of the Internet", Communications of the ACM, November 1994 - Volume 37, Number 11, Page 18

¹³ Information Technology Security Evaluation Criteria (ITSEC), auch White Book genannt, wurde von der deutschen Bundesanstalt für sichere Informationstechniken (BSI) im Zuge einer internationalen Standardisierung im Bereich Rechnersicherheit herausgegeben.

Datenmanagement

Mit der Verteilung des Systems stellt sich natürlich auch die Frage, wie die notwendigen Daten gehalten werden. Um identische Daten an unterschiedlichen Lokationen gleichzeitig verfügbar zu haben, gibt es zwei Möglichkeiten. Entweder werden die Daten an jeder Stelle lokal gehalten, wodurch sich die Problematik stellt, wie Datenbestände bei Änderungen konsistent gehalten werden können. Oder aber die Daten werden systemintern durchgereicht, wodurch einerseits große Abhängigkeiten von Teilsystemen, andererseits Zeitverluste aufgrund von internen Kommunikationen entstehen. Beim Verkauf von Produkten können grob zwei Arten von Daten unterschieden werden, die produktspezifischen Daten, und die personenspezifischen Daten.

Bei den produktspezifischen Daten handelt es sich im allgemeinen um große Datenpakete und deren Beschreibungen. Normalerweise werden diese Produkte in jeder Filiale angeboten. Diese Daten müssen in der Regel nur solange online verfügbar sein, wie das Produkt angeboten wird. Da es sich um urheberrechtlich geschützte Daten handelt, dürfen sie nicht für Dritte zugänglich sein. Diese großen Datenvolumen werden sinnvollerweise in jedem Knoten lokal gehalten. Die einzelnen Filialen sind somit bei der Auslieferung der Produkte eigenständig. Begleitinformation zu den einzelnen Produkten kann leider nicht in diesem Maße entkoppelt werden. Dies bedeutet, daß innerhalb kürzester Zeit alle Filialen erreichbar sein müssen, um diese Informationen, z.B. Preise, zu aktualisieren. Da es sich hierbei um geringe Datenvolumen handelt die häufig benutzt aber nur selten geändert werden, werden diese Daten lokal gehalten.

Bei den personenspezifischen Daten wird unterschieden zwischen Anbieter und Käufer. Anbieterdaten werden vor allem zu Abrechnungszwecken benötigt. Sie werden ebenfalls lokal gehalten, müssen jedoch bei jedem Verkauf aktualisiert werden. Da jeder Kunde die Möglichkeit haben soll, in jeder Filiale einzukaufen, und seine individuellen Daten abzurufen bzw. zu aktualisieren, müssen zumindest Teile der kundenspezifischen Daten innerhalb möglichst kurzer Zeit von jedem Knoten aus systemweit eindeutig verfügbar sein. Zur *Beweissicherung* werden alle Benutzeraktionen mitprotokolliert und enthalten somit natürlich auch alle durchgeführten Transaktionen. Es kommt daher zu einem großen Kommunikationsbedarf von sensiblen Informationen geringen Datenvolumens, wobei sich Wartezeiten negativ bemerkbar machen. Sind die kundenspezifischen Daten, beispielsweise Kontingente, zu einem Zeitpunkt innerhalb des gesamten Systems nicht identisch, so kann dies finanziell fatale Folgen haben. Aus diesem Grunde wird hier trotz der Zeitanforderungen das Durchreichen von Information interessant, um die Eindeutigkeit der Information garantieren zu können.

Gewährleistung der Funktionalität

Wie können die zwei konträren Anforderungen erfüllt werden: Untergliederung des Gesamtsystemes in weitestmöglich autonome Einheiten und Einkaufsmöglichkeit jedes Benutzers in jeder Filiale? Wie wird das System konzipiert um eine möglichst große Gewährleistung seiner Funktionalität bieten zu können?

Die Synchronisation der lokal gehaltenen produktspezifischen Daten wird systemintern durchgeführt. Zielsetzung ist, daß entsprechend der Vertriebsordnung alle Filialen ein möglichst identisches Angebot aufweisen sollen. Enthalten die Filialen kurzzeitig ein nicht 100 Prozent gleiches Angebot, so ergibt sich hieraus kein größerer Schaden. Beim Angebot ist strengstens darauf zu achten, daß sowohl die produktbezogenen Daten als auch die Produkte selbst verfügbar sein müssen, bevor sie zum Kauf freigegeben werden. Dringliche Informationen, wie Sperren eines Produktes, können über eine Administrationsschnittstelle erfolgen. Mit diesem Programm können von unterschiedlichen Lokationen aus, und je nach Authorisierung, verschiedene Aktionen auf einzelnen Teilsystemen ausgeführt werden. Falls diese Administration einer Filiale nicht mehr garantiert werden kann, wird diese automatisch geschlossen.

Die Handhabung der kundenbezogenen Daten stellt die größten Anforderungen an das System. Jeder Kunde wird einem Teilsystem zugeordnet. Möchte ein Kunde die Möglichkeit haben, in jeder Filiale einzukaufen zu können, so würde dies bedeuten, daß jedes Teilsystem mit jedem Teilsystem kommunizieren können muß. Zur Koordination der Kommunikation unter den Filialen wird deshalb eine zentrale Einheit eingeführt, in welcher alle Benutzer verzeichnet sind. Solange die Verbindung zur Vermittlungsstelle und zwischen der Vermittlungsstelle und der Filiale existiert, in welcher sich alle Daten des Kunden befinden, kann der Kunde in jeder beliebigen Filiale einzukaufen. Die kundenspezifischen Daten werden durchgereicht. Diese Vermittlung ermöglicht auch die Kontrolle, daß jeder Benutzer zu jedem Zeitpunkt nur einen Einkaufsvorgang durchführen kann. Jede Neuregistrierung erfolgt grundsätzlich über die Zentrale. Als Lokation der Kundendaten wird sinnvollerweise das Teilsystem gewählt, welches von der Netzverbindung am geeignetsten ist, dies wird im allgemeinen landesintern sein.

Diese zentrale Komponente bildet in dem Gesamtsystem natürlich einen Engpaß. An dieser Stelle wird aus Gründen der Komplexität dieser Grad an *Fehlerüberbrückung* bzw. *Gewährleistung der Funktionalität des Systems* zunächst als ausreichend bezeichnet und verstärkt auf weitere Sicherheitskriterien wie Authentifikation, Rechteprüfung, Rechteverwaltung und vor allem auf die Übertragungssicherung eingegangen.

Sicherheit

Sicherheit und Benutzerfreundlichkeit sind zwei Themen welche an vielen Stellen stark entgegenwirken. Einerseits bedeutet Sicherheit häufig zusätzliche Aktionen von Seiten des Benutzers, andererseits können durch systeminterne Abläufe auch Wartezeiten entstehen, welche für den Benutzer unangenehm sind. Auch in diesem System, erfolgt die *Identifikation* und *Authentifikation* über eine Benutzeraktion. Um dem Benutzer zumindest das Auswendiglernen fremder Zeichenfolgen zu ersparen, hat der Benutzer die Möglichkeit, sich sowohl den Identifikator als auch den Authentifikator selbst zu wählen. Wartezeiten durch sicherheitstechnische Aktionen entstehen bei der Rechteprüfung und bei der Übertragungssicherung. Verzögerungen bei der Rechteprüfung werden beim Einsatz schneller Rechner auf Dienstseite vernachlässigbar gering.

Bei der *gesicherten Datenübertragung* wird unterschieden zwischen symmetrischen und asymmetrischen Verschlüsselungsverfahren. Dabei sind die symmetrischen Verfahren im allgemeinen die Effizienteren. Sie setzen jedoch voraus, daß beide Kommunikationspartner denselben nur ihnen bekannten Schlüssel besitzen. Hierbei stellt sich das Problem wie die Schlüssel noch vor Beginn der eigentlichen gesicherten Datenübertragung ohne Kenntnis Dritter vereinbart werden können. Bei asymmetrischen Verfahren wird zwischen öffentlichen und privaten Schlüsseln unterschieden. Der Vorteil hierbei ist, daß nur öffentliche Schlüssel bekannt gegeben werden müssen. Durch die Kombination von asymmetrischen wie symmetrischen Mechanismen, lassen sich die Vorteile beider Ansätze vereinen.

Dabei wird ein asymmetrisches Schlüsselaustauschverfahren gewählt, bei welchem die Schlüssel zur Laufzeit erzeugt werden. Mithilfe dieses Basismechanismus lassen sich dann auch gesichert Schlüssel für das symmetrische Verfahren aushandeln. Durch die Veränderung der Anzahl der Datenworte, die mit demselben Schlüssel codiert werden, läßt sich das Verhältnis zwischen Schlüssel und Nutzdaten beeinflussen. Hierbei stehen der Grad an Sicherheit und die Effizienz des Verfahrens in umgekehrten Verhältnis zueinander.

Effizienz

Selbst bei der Auswahl von sehr effizienten Verschlüsselungsmechanismen kommt es zu Effizienzeinbußen durch die gesicherte Übertragung. Diese Effizienzeinbußen ergeben sich einerseits durch das Faktum, daß mehr Daten übertragen werden müssen, andererseits, dadurch daß das Aushandeln der Schlüssel vollständig abgeschlossen sein muß, bevor die eigentliche Kommunikation stattfinden kann.

Um die Antwortzeiten bei Benutzeranfragen klein zu halten, werden mehrere Anfragen simultan bearbeitet. Eine Effizienzsteigerung läßt sich hierbei durch die gegenseitige Überlagerung von Berechnungen, Kommunikation und Datenbankabfragen erreichen. Dies kann prinzipiell durch das Abspalten von Prozessen für einzelne Benutzeranfragen geschehen. Vorteilhafter ist hier jedoch die Verwendung von leichtgewichtigen Prozessen zumal dies die einfache Kommunikation der einzelnen Aktivitäten über gemeinsame Daten erlaubt.

Im Rahmen von Effizienz sind natürlich auch Betrachtungen interessant, wie eventuell auftretende Wartezeiten durch entsprechende Benutzerführung überbrückt werden können. Aktionen, welche nicht unmittelbar im Mittelpunkt des Benutzerinteresses stehen, können nebenläufig ausgeführt werden. Beispielsweise können Standardpreise immer unmittelbar angezeigt werden, während die benutzerspezifischen Preise, aufgrund von Sonderkonditionen automatisch nachgeliefert werden.

Administration

Auch im Rahmen systeminterner Aktionen, wie der Administration, ist Sicherheit ein nicht zu vernachlässigendes Thema. Bei sensiblen Daten stellen sich immer die Fragen: Wer darf welche Information einsehen oder verändern? Wer darf welche Aktionen ausführen? Hier ist ganz klar zwischen reinen Wartungsaktivitäten des Systems selbst, und zwischen anwendungsbezogenen Verwaltungsaktivitäten zu unterscheiden. Innerhalb des verwaltungstechnischen Bereichs ist es des weiteren sinnvoll

nach Aktionen, wie beispielsweise Registrierung oder Rechnungserstellung, zu untergliedern. Außerdem sollten weitmöglichst klare Zuständigkeiten für abgrenzbare Datenbereiche gewählt werden. Auch hier ist es interessant, daß die einzelnen Filialen soweit wie möglich autonom behandelt werden. Bei der *Rechteverwaltung* ist somit eine Lösung, bei der die Zugriffsrechte quasi frei zwischen Anwendern und zu schützenden Objekten zugeordnet werden, wie dies in der diskreten Sicherheitspolitik der Fall ist, von Interesse. Eine Anbindung der Rechte an die Aktionen ist hierbei sinnvoll. Die Rechteverwaltung selbst wird über höhere Instanzen ausgeführt. Auch hier wird genau festgelegt, wer Rechtezuordnungen erzeugen, ändern und löschen darf.

Ein grundsätzlicher Aspekt ist, die Übersicht über die Auswirkungen der ausgeführten Aktionen zu bewahren. Aktionen, welche keine Veränderungen bewirken, können von beliebig vielen Anwendern gleichzeitig ausgeführt werden. Handelt es sich allerdings um verändernde Aktionen, muß genau festgelegt werden, wer parallel zu dieser Aktion Zugriff haben darf. Bei verändernden Aktionen werden deshalb temporär die Rechte anderer eingeschränkt. Dies ermöglicht das Sicherstellen, daß egal welche Aktion ausgeführt wird, zu jedem Zeitpunkt der Datenbereich konsistent erhalten bleibt.

Schutz vor *Wiederaufbereitung* ist ein Sicherheitsaspekt, der durch die viel alltäglichere Art der Verwaltung, wie Account-Vergabe und Gebäudezugangsbeschränkungen nur für vertrauliches Personal, in Angriff genommen werden kann.

Anpaßung an sich ändernde Anforderungen

Für den kompletten elektronischen Vertrieb ist ein sehr breites Spektrum an zu haltenden Daten gegeben. Dieses Spektrum reicht von den produktspezifischen Daten, wie Kurzbeschreibungen, Verfügbarkeit, Randbedingungen, Preise, der Querverbindung zu den vertraglichen Informationen, wie Eigentümeradressen und Konditionen, über den Kunden mit seinen Registrations- und Transaktionsdaten, bis schlußendlich zu landesspezifischen Daten wie Wechselkursverwaltung. Zur Haltung dieser Daten kommen Datenbanken mit entsprechender Zugangsbeschränkung zum Einsatz.

Um an den einzelnen Knoten eventuell existierende Datenbanklizenzen ausnutzen zu können, wurde bei der Erstellung der Datenbank nicht die endgültige Auswahl für ein bestimmtes Datenbanksystem getroffen. Vielmehr wurde die Datenbankschnittstelle soweit abstrahiert, daß verschiedene Datenbanksysteme gleichzeitig zum Einsatz kommen können.

Bei der großen Vielfalt unterschiedlicher Daten ist die Wahrscheinlichkeit, daß sich die Anforderungen mehrfach ändern, absehbar. Da Änderungen von zu haltenden Daten zwangsläufig bis in jede Datenbank durchdringen, ist es sehr wichtig, sich hier schon von Anfang an auf eine möglichst dynamische Lösung vorzubereiten. Hierfür bieten sich Generatoren an. Für jedes Datenbanksystem muß allerdings zunächst ein Schema geschrieben werden. Da sich Änderungen von zu haltenden Daten natürlich auch auf die Kommunikation und Administration auswirken, ist es interessant, auch Protokolle und die Administrationsschnittstelle weitmöglichst semi automatisch zu erzeugen.

Projekte

Mobile Community & Workflow Initiative (MCW)

Stefan Gessler

1 Das Konsortium

Die Einführung portabler Computer, weltweit zugreifbare Rechnernetze und die wachsende Mobilität in der Arbeitswelt führt zu der Frage, wie die fortschreitende Technologie helfen kann, die tägliche Arbeit zu organisieren. Speziell werden auch die sich immer mehr verbreitenden Workflowsysteme durch die wachsende Mobilität betroffen sein.

Mit der *Mobile Community & Workflow Initiative (MCW)* haben das CEC Karlsruhe, die Institute für Informatik an den Universitäten Darmstadt und Dresden, sowie das Telecooperation Office ein Konsortium gegründet, dessen Aufgabe es ist, ein allgemeines Mobilitätsmodell und eine allgemeine Architektur für ein Dienstsysteem zu entwickeln, das eine Integration mobiler Benutzer in vorhandene Workflow Systeme und die Erweiterung der Gültigkeit eines Workflowsystems über über lokale Netze hinaus auf Weitverkehrsnetze ermöglicht.

2 Motivation

In unserer Informationsgesellschaft ist die rechentechnische Unterstützung von Büroabläufen in der industriellen Praxis eine wesentliche Voraussetzung für Effizienz, Wirtschaftlichkeit und Konkurrenzfähigkeit. Durch zunehmend dezentrale, weiträumig verteilte und gleichzeitig hochdynamische Organisationsstrukturen gewinnt dabei die Mobilität der beteiligten Mitarbeiter rapide an Bedeutung. Konkrete Beispiele hierfür sind in den meisten Industriebetriebe anzutreffen: Sachbearbeiter, Techniker und Ingenieure sind zwischen vielen verschiedenen Einsatzorten im In- und Ausland unterwegs, von handwerklichen Tätigkeiten bis hin zu industriellen Großprojekten

Dabei müssen - unter Nutzung mobiler Rechner - ständig umfangreiche digitalisierte Informationen verarbeitet werden, so etwa zu aktuellen Projekt- und Logistikdaten oder zu betriebswirtschaftlichen Fragestellungen, also zu umfangreichen organisatorischen und technischen Prozessen im Rahmen komplexer Gesamtsysteme. Gleichzeitig ist häufig ein weiträumiger Zugriff auf zentralisierte oder auch gezielt verteilte Informationsbestände über moderne Rechnernetze erforderlich. Die notwendige systemtechnische und Hardware-orientierte Unterstützung für solche Zielstellungen ist inzwischen weit entwickelt: Mit mobilen Rechnern (v.a. Notebooks und Notepads), mobilen Kommunikationsnetzen (speziell GSM, Modacom und lokale Funknetze) sowie elementaren Kommunikationsprotokollen (z.B. Mobile-IP) stehen wesentliche Basiskomponenten für fortgeschrittene mobile Anwendungen bereit. Diese Bereiche stellen aber lediglich eine erste Grundlage dar; entscheidende zusätzliche Anforderungen und neue Herausforderungen für Forschung und Entwicklung stellt die geeignete softwaretechnische Unterstützung, die daher Inhalt der konsortialen Arbeiten ist.

Die wesentlichen Anforderungen lassen sich wie folgt gliedern.

- Einfacher, effizienter und kostengünstiger Datenzugriff durch mobile Stationen
- Explizite Modellierung organisatorischer/betrieblicher Abläufe (Workflows)
- Gezielte Unterstützung von Workflows in mobilen Umgebungen
- Entwurfstechniken und -werkzeuge für mobile Anwendungen
- Beherrschung der dynamischen Konfiguration mobiler Systeme
- Fortgeschrittene Techniken der Ressourcenverwaltung beim mobilen Zugriff
- Systematische Unterstützung durch eine integrierte Gesamtarchitektur
- Integration aktueller Mobilnetze in das Gesamtkonzept
- Zukunftssicherheit durch Aufsetzen auf internationalen Standards
- Experimentelle Validierung und Praxiseinsatz durch konkrete Anwender

Die genaue Analyse dieser Anforderungen zeigte, daß die heutigen Systeme bestenfalls erste rudimentäre Lösungen für die ersten beiden Bereiche bieten, die aber für die fortgeschrittenen Anwendungen bei modernen Unternehmen unzureichend sind.

3 Aufgabe des Konsortiums

Im Rahmen des Konsortiums soll eine integrierte, innovative Software-Architektur zur Unterstützung mobiler Rechneranwendungen in der industriellen Praxis geschaffen werden. Dabei werden insbesondere objektorientierte Ansätze gewählt, die eine klare Strukturierung der zu entwickelnden Software in dedizierte Bausteinbibliotheken unterstützen und die anwendungsübergreifende Wiederverwendung von Softwarekomponenten stark fördern. Im einzelnen gliedert sich das Vorhaben in die folgenden Teilbereiche der verschiedenen Partner:

3.1 System und Benutzerverwaltung (TU Dresden)

Dieser Teilbereich umfaßt die Entwicklung der erforderlichen Basis-Softwareinfrastruktur für mobile Anwendungen. Neben einer geeigneten Strukturierung hochdynamischer mobiler Systeme wird die Verwaltung der dynamischen Konfiguration, die Lokalisierung von Ressourcen durch Anwendungen, die gezielte Planung von Kommunikationsvorgängen und die systematische Unterstützung mobiler Nutzer in wechselnden Einsatzumgebungen im Rahmen einer umfangreichen Bausteinbibliothek realisiert.

3.2 Modellierungssystem für mobile Anwendungen (Telecooperation Office)

Die elementaren Basiskomponenten werden durch eine entsprechende Bausteinbibliothek zur Realisierung konkreter mobiler Anwendungen unterstützt. Dabei werden vordefinierte Komponenten zur Konfigurierung von Applikationen angeboten, wobei ganz besonders die Aspekte der mobilen Kommunikation und der Dynamik mobiler Komponenten berücksichtigt werden. Zur Unterstützung des schrittweisen Entwurfs wird ein ergänzendes Modellierungssystem entwickelt, das insbesondere auch speziell auf die mobile Umgebung abgestimmte Entwurfswerkzeuge umfaßt.

3.3 Werkzeuge und Methoden für mobile Vorgangsbearbeitung (CEC Karlsruhe, TH Darmstadt)

Für das vorgegebene Anwendungsfeld spielen mobile Vorgänge (Workflows) eine ganz besondere Rolle. Unter Nutzung der obenbeschriebenen Komponenten zur Systemstrukturierung und zur Entwicklung mobiler Anwendungen werden daher spezielle Techniken zur Unterstützung der mobilen Vorgangsbearbeitung entwickelt. Neben der formalen Vorgangsbeschreibung spielen dabei vor allem der Zugriff auf Ressourcen und Daten in mobilen Umgebungen sowie die dynamische An- und Abkopplung von Bearbeitern eine wesentliche Rolle.

4 Beteiligung des Telecooperation Office

Die Aufgabe des TecO ist es, den Entwickler mobiler Anwendungen beim Entwurf zu unterstützen. Hierzu werden eine Bausteinbibliothek und ein Modellierungssystem für mobile Anwendungen realisiert. Dieses Vorhaben ist mit den Arbeiten der Partner eng verzahnt: Die softwaretechnische Systemunterstützung realisiert die Technische Universität Dresden durch eine Bausteinbibliothek für die System- und Benutzerverwaltung in mobilen Umgebungen und stellt damit die Basis für die Arbeiten der Universität Karlsruhe zur Verfügung. Das CEC entwickelt gemeinsam mit TH Darmstadt dann darauf abgestimmt dedizierte softwaretechnische Werkzeuge und Methoden für die mobile Vorgangsbearbeitung unter Nutzung der Dresdner Systemkomponenten. Durch die Entwicklung einer formalen Vorgangsbeschreibung unter spezieller Berücksichtigung abgekoppelter und mobiler Datenzugriffe wird gleichzeitig ein sehr allgemeingültiger, universell nutzbarer Ansatz erzielt.

Bei den Forschungs- und Entwicklungsaufgaben des TecO stehen folgende Zielsetzungen im Vordergrund.

4.1 Dynamik mobiler Anwendungen

Ein grundlegendes Problem bei der Entwicklung mobiler Anwendungen ist die inhärente Dynamik. Hierzu wird insbesondere die mobilitätsspezifische Datenabhängigkeit von Anwendungen untersucht. Es wird ein Datenmodell erstellt, durch das solche Abhängigkeiten beschreibbar sind. Außerdem wird untersucht, in welcher Art dem Anwendungsprogrammierer Funktionsunterstützung bei anwendungsnahe Daten- und Ressourcenzugriff angeboten werden kann. Hier spielt insbesondere das Kommunikationssystem eine bedeutende Rolle. Desweiteren wird ein Modell entwickelt, durch das die Abbildung eines abstrakten Ressourcenzugriffs auf eine reale Ressource spezifiziert werden kann. Schließlich wird eine Spezifikationsmöglichkeit der Behandlung/Reaktion bei abgekoppelten, replizierten Daten geschaffen.

4.2 Modellierungssystem und Entwurfsunterstützung

Auf der Grundlage der Untersuchungen und Modellbildungen zur Dynamik mobiler Anwendungen wird ein Modellierungssystem erstellt. Dies umfaßt insbesondere eine Bausteinbibliothek, die mobilitätsspezifische Aspekte mobiler Anwendungen realisiert und in Form einer Programmierschnittstelle angeboten wird. Zusätzlich wird ein graphisches Modellierungswerkzeug realisiert, mit Hilfe dessen das Datenmodell für eine mobile Anwendung entworfen werden kann. Der Anwendungsprogrammierer spezifiziert so unter anderem interaktiv, wo Daten gehalten werden und welche Daten für die Ausführung unbedingt erforderlich sind.

4.3 Benutzerfreundliche Interaktionsformen

Letztlich ausschlaggebend für den Erfolg mobiler Anwendungen ist die Akzeptanz beim Benutzer. Diese hängt entscheidend von der angebotenen Benutzerschnittstelle ab. Aktuelle Entwicklungen im Bereich Mensch-Maschine-Schnittstellen haben multimodale Benutzerschnittstellen zum Gegenstand, bei denen der Anwender mehrere Eingabemöglichkeiten (Tastatur, Maus, Sprache, Gestik etc.) parallel oder alternativ nutzen kann, wodurch eine wesentlich intuitiver Bedienung erreicht wird. Im Rahmen des Vorhabens wird untersucht, inwieweit modalitätsunabhängige Interaktionsformen im mobilen Umfeld einsetzbar sind. Die Beschreibungsmodelle werden entsprechend der Ergebnisse erweitert, und die Bausteinbibliothek wird um Komponenten zur Spezifikation modalitätsunabhängiger Interaktionsformen ergänzt. Schließlich wird auch das Modellierungswerkzeug um die notwendigen Funktionalitäten erweitert.

5 Durchführung

Die beschriebenen Entwicklungen sollen durch den Einsatz bei Pilotpartnern validiert werden. Dabei spielt speziell die Unterstützung mobiler Sachbearbeiter bei der Bearbeitung organisatorisch/technischer Vorgänge eine zentrale Rolle. Gleichzeitig wird aber auf Adaptierbarkeit aller Konzepte und Systemlösungen auf andere Anwendungsfelder geachtet. Insgesamt deckt das Konsortium alle wesentlichen Bereiche der Unterstützung mobiler Rechneranwendungen durch Software-Komponenten ab; dies umfaßt die elementare Systemunterstützung, den Entwurf und die Modellierung der Anwendungen, die gezielte Unterstützung betrieblicher Abläufe sowie die anwenderbezogene Validierung der Konzepte.

Das Projekt hat bezüglich der Ausrichtung zukünftiger Betätigungsfelder der Partner strategische Bedeutung. Daher ist der Fokus nicht auf kurzfristig verfügbare Systeme, sondern mittelfristig auf eine umfassende Behandlung aller relevanten Aspekte ausgerichtet.

Multimedia Collaboration

Jörg Sievert

1 Einführung

Das hier vorgestellte Drittmittelprojekt wird im Rahmen des BERKOM-Projekts entwickelt[MMC93]. BERKOM ist ein breit angelegtes Forschungsprojekt der DeTeBerkom mit Beteiligung mehrerer Unternehmen, um Breitbandtechnologien zu entwickeln und zu testen. Die erste Phase von BERKOM endete 1991. Ihr Ziel war die Untersuchung vieler unabhängiger Breitbandnetzwerkösungen. Die zweite Phase, welche 1992 begann und 1994 endete, konzentrierte sich auf eine einzelne Kommunikationsinfrastruktur, mit der Multimedia-Anwendungen über ein Breitbandnetz aufgebaut werden können. In der zweiten Phase wurden vier Forschungsthemen betrachtet. Als Basis wurde ein *Multimedia-Transportsystem* entwickelt, welches die Qualitätsanforderungen einer multimedialen Anwendung erfüllen kann. Darauf aufbauend wurde ein *Multimedia-Mail-System* entwickelt, als Erweiterung des X.400 Mail-Systems, welches die transparente Versendung von multimedialen Briefen erlaubt. Verbunden hiermit ist die Entwicklung eines *Multimedia-Archivs* im Netz. Für diesen Bericht interessant ist vor allem die vierte Anwendung, das *Multimedia-Collaboration-System*[ADH⁺92].

2 Multimedia-Collaboration-System

Das Multimedia-Collaboration-System realisiert Konferenzen zwischen entfernten Teilnehmern. Die Teilnehmer kommunizieren dazu durch Mikrofon, Kopfhörer und Videokamera miteinander. Die Video- und Audiodaten werden vom System digitalisiert und zwischen den Teilnehmern übertragen. Während die Videoströme einzeln dargestellt werden, werden die einzelnen Audiodatenströme gemischt, sodaß der Eindruck eines Gesprächs in einem Raum erzeugt wird. Die Teilnehmer können außerdem in die Konferenz jede beliebige X-Windows-Anwendung einbringen. Durch MMC können alle Teilnehmern diese Anwendungen sehen und bedienen. MMC koordiniert dabei die Eingaben der Teilnehmer anhand zweier Kooperationsmuster. Im *expliziten Modus* liegt die Kontrolle der Anwendung bei einem Teilnehmer. Andere können die Kontrolle explizit anfordern und der kontrollierende Teilnehmer kann die Kontrolle explizit abgeben. Im *impliziten Modus* führt eine kurze Inaktivität des kontrollierenden Teilnehmers zur Weitergabe der Kontrolle an einen aktiven Anwender. Aus dem System ist dabei ersichtlich, wer gerade die Kontrolle innehat. Die Teilnehmer müssen sich dann über den Audio-Kanal in der Benutzung der Anwendung koordinieren.

MMC ist in folgende Komponenten strukturiert (siehe Abbildung 43):

- ASC** Die Application-Sharing-Komponente fügt sich in das X-Windows-System ein und übernimmt die Verteilung der Oberfläche der Anwendungen, die in der Konferenz gemeinsam betrachtet werden. Sie übernimmt auch die Verteilung der Eingaben an die Anwendung.
- AVC** Die Audiovisual-Komponente stellt die Audio- und Videodaten auf dem lokalen Rechner eines Teilnehmers dar und sammelt die entsprechenden lokalen Daten. Sie benutzt dazu die jeweils verfügbare Hardware. Sie steht mit der AVC der anderen Teilnehmer in Verbindung.
- AVM** Der Audiovisual-Manager koordiniert die Aushandlung von Qualitätsmerkmalen für die Audio- und Videokanäle zwischen den Teilnehmern.
- CD** Das Conference-Directory stellt eine zentrale Datenbasis für Benutzer, Benutzergruppen und Konferenzen dar.
- CIA** Der Conference-Interface-Agent ist die Schnittstelle des Teilnehmers zum MMC-System.
- CM** Der Conference-Manager ist der Kern des MMC. Er verwaltet und koordiniert Konferenzen mit Hilfe der anderen Komponenten.
- IB** Der Invitation-Broker empfängt für Teilnehmer Einladungen zu Konferenzen und leitet diese an die entsprechenden Teilnehmer weiter.

Zwischen den Komponenten werden folgende Protokolle benutzt:

- Der Conference-Manager und die Conference-Interface-Agents kommunizieren über das *Conference-Directory-Access* (CDAP) Protokoll mit dem Conference-Directory.
- Der Conference-Manager kommuniziert mit den Conference-Interface-Agents über das *Conference-Manager-Access* (CMAP) Protokoll.

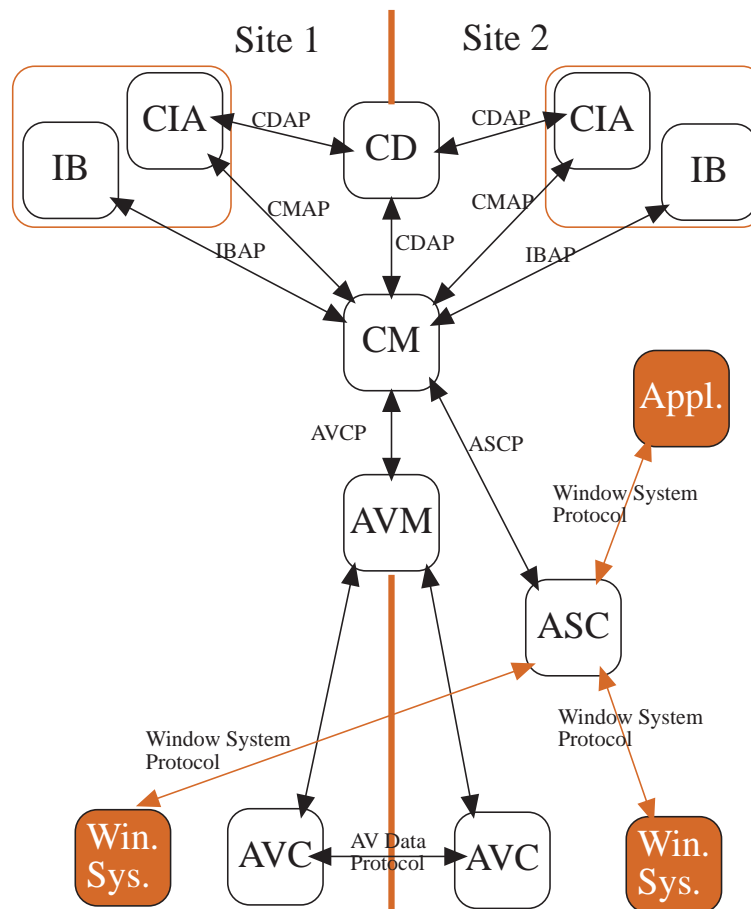


Abbildung 43. MMC Komponenten und Protokolle

- Der Conference-Manager kontrolliert die Application-Sharing-Komponente über das *Application-Sharing-Control* (ASCP) Protokoll.
- Den Audiovisual-Manager kontrolliert der Conference-Manager mit dem *Audiovisual-Control* (AVCP) Protokoll.
- Der Audiovisual-Manager kommuniziert mit allen Audiovisual-Komponenten über das *Source-and-Sink-Control* (SSCP) Protokoll.

3 Arbeiten am Institut

Im Jahr 1994 war das Institut für Telematik in Kooperation mit dem DEC-CEC an folgenden Komponenten mit einer Reihe von Arbeiten beteiligt.

- Conference-Manager
Der Conference-Manager als zentrales Element innerhalb der Komponenten von MMC wurde auf Basis von ISODE weiterentwickelt.
- Conference-Directory
Die bisherige Entwicklung des Conference-Directory mit Quipu bzw. in einer zweiten Version auf Textdateibasis wurde durch eine vollkommene Neuentwicklung auf der Basis von DEC's X.500 Implementierung ersetzt.
- Conference-Interface-Agent
Die Entwicklung der Oberfläche, welche der Benutzer zum Konferenzsystem hat, wurde in drei Teile aufgebrochen, um unabhängig vom Kommunikationssystem zu sein. Derzeit basiert MMC noch auf ISODE, es wird aber geplant, es durch OSF/DCE zu ersetzen.
- Invitation-Broker
Der Invitation-Broker wurde in einer eigenen Entwicklung abweichend vom Projektstandard in seiner Funktionalität erweitert und behandelt mehrere Benutzer. Außerdem wurden elementare Sicherheitsaspekte hinzugefügt.

Mensch-Maschine Interface im multimedialen Verwaltungsbüro

Sonja Zwißler

Im Bereich der öffentlichen Verwaltung stellen immer komplizierter und spezieller werdende Aufgaben und immer kürzer werdende Zeiträume zur Absprache und Entscheidungsfindung wachsende Herausforderungen an die Organisation der Arbeitsprozesse. Der Beschluß des Deutschen Bundestages zur Vollendung der Deutschen Einheit führt zu einer Verteilung der Verfassungsorgane zwischen Berlin und Bonn. Durch diese zunehmende Dezentralisierung der Ämter wird die Kommunikation immer dringlicher um gemeinsame Aufgabenziele auch über große geographische Entfernungen hinweg zu erreichen. Im Zuge der Europäischen Union wird die Kooperation der Verwaltungsämter auch international erforderlich sein. Leistungsorientierung, schlanke Verwaltung oder Outsourcing sind Schlagwörter, die auch in der öffentlichen Verwaltung Einzug halten. Mit neuen Techniken aus dem Bereich der Telekooperation wird es möglich, die notwendigen komplexen informationstechnischen Strukturen zu realisieren und den Menschen bei der Erfüllung seiner Tätigkeiten wirkungsvoll zu unterstützen. Diese Dienste werden im Rahmen dieses Verbundprojektes, unter der Leitung der DeTeBerkom mithilfe einer benutzerfreundlichen Mensch-Maschine Schnittstelle dem Anwender nähergebracht.

Im folgenden wird einführend auf die Zielsetzung des Projektes eingegangen. Nach einer Beschreibung des Gesamtprojektes werden dann die TecO spezifischen Arbeiten vorgestellt. Abschließend werden die Projektdurchführung und dessen Organisation angesprochen.

1 Zielsetzung des Verbundprojektes

Zielsetzung des Projektes ist die Unterstützung der öffentlichen Verwaltungen im Anwendungsbereich Bürokommunikation, durch den Einsatz und die Nutzung von Telekooperationsanwendungen. Grundvoraussetzung hierfür ist die Zurverfügungstellung von Anwendungen über möglichst benutzerfreundliche Schnittstellen. Das Projekt **Mensch-Maschine Interface im multimedialen Verwaltungsbüro** (MMI) soll dabei der Entwicklung, Pilotierung und Bewertung neuer MMI-Verfahren für eine Verbesserung der Oberflächengestaltung für Telekooperationsanwendungen dienen.

2 Beschreibung des Verbundprojektes

In der Studie "IT-Unterstützung im Informationsverbund - Berlin-Bonn" (IVBB) werden Aufgabenfelder bzw. Geschäftsvorfälle innerhalb der Regierungs- und Verwaltungstätigkeit extrahiert. Aus ihnen werden in der Studie die folgenden Kommunikationsvorgänge abgeleitet: persönlich besprechen, telefonieren, Dokumentenaustausch (Dokumente empfangen, senden und verteilen) und elektronische Informationsgewinnung und -speicherung (Datenbanken, Archive, Verfahrensnutzung). Viele Aufgaben in der öffentlichen Verwaltung werden heute bereits mit Hilfe des Computers bearbeitet. Das Ergebnis der Arbeiten, beispielsweise ein Dokument, liegt jedoch zumeist als Ausdruck auf Papier vor. Wenn es darum geht, ein Dokument von einer Person überarbeiten zu lassen, die sich zur Zeit nicht vor Ort befindet, kommt die herkömmliche Post zum Einsatz. Um diese Papierflut zu vermeiden, und vor allem bei schnellen Entscheidungen bietet sich der Einsatz eines computerunterstützten Werkzeuges an, das die andere Person befähigt, etwa direkt in einer Telekonferenz das Dokument zu bearbeiten. Unterstützt von audiovisueller Kommunikation lassen sich so innerhalb kürzester Zeit, trotz großer Entfernung der verschiedenen beteiligten Personen, Aufgaben effizienter durchführen.

Die Verwirklichung des multimedialen Verwaltungsbüros setzt neben der Realisierung von komplexen Telediensten eine software-ergonomische Gestaltung der Benutzungsschnittstelle voraus. Als Verbindung zwischen dem arbeitenden Menschen und dem für die Arbeit benutzten System bestimmt sie nicht nur die Akzeptanz des Systems und die Effizienz seines Einsatzes, sondern auch die menschengerechte Gestaltung der Arbeit beim Umgang mit dem Rechner. Durch das breite Spektrum an Benutzern, vom Computerneuling bis hin zum Informatikfachpersonal, vom Dienstpersonal bis in die Führungsebene werden sehr unterschiedliche Anforderungen an die Bedienung der Schnittstelle gestellt.

Um über diese Benutzungsschnittstelle einerseits die Nutzung der verschiedenen kooperativen Technologien zu ermöglichen und andererseits diese Nutzung möglichst komfortabel und benutzeradäquat zu gestalten, wird ein System entworfen, bei welchem die folgenden Möglichkeiten genutzt werden. Als kooperative Technologien bieten sich die Berkom Teledienste: Multimedia Collaboration (MMC) und Multimedia Mail (MMM) an. Eine möglichst komfortable und benutzeradäquate Gestaltung der Schnittstelle kann einerseits durch den Einsatz von Interaktionsgeräten erfolgen, die sich für die entsprechenden Anforderungen eignen. Hier bietet sich beispielsweise ein Penboard für Gruppenarbeit, ein

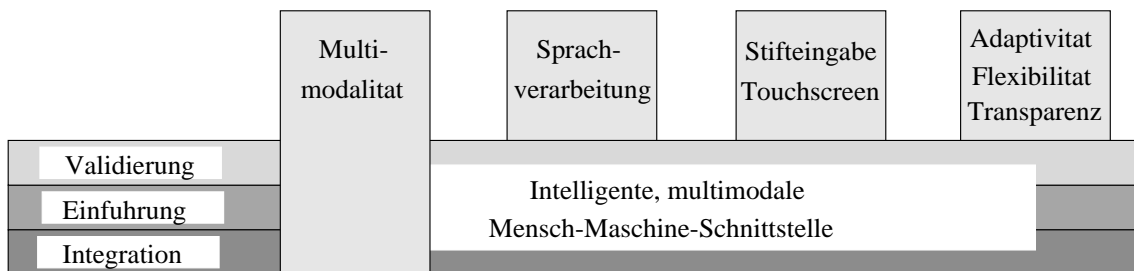


Abbildung 44. Aufgabenstruktur der MMI-Dienste

Stift für Formulardatenerfassung oder aber ein Touchscreen für Informationskioske an. Andererseits kann eine möglichst komfortable und benutzeradäquate Gestaltung der Schnittstelle durch Softwaretechnologien erfolgen. Dazu zählt beispielsweise die Möglichkeit der Adaption einer Schnittstelle an benutzerspezifische Bedürfnisse. Eine flexible und transparente Benutzerführung erlaubt dem Anwender gewisse Freiheitsgrade für die Gestaltung seines Arbeitsablaufs. Multimodalität bedeutet den Einsatz verschiedener Modalitäten.

Entsprechend dieser Möglichkeiten wird innerhalb dieses Projektes ein MMI-System bestehend aus physikalischen und software-technischen Diensten erstellt. Zu den physikalischen Diensten zählen neben der Verfügbarkeit der herkömmlichen Dialoggeräte: Bildschirm, Tastatur und Maus die Dialoggeräte: Penboard, Stift, Touchscreen und die Sprachverarbeitung. Zu den software-technischen Diensten gehören der Adaptivitätsdienst, der Flexibilität/Transparenzdienst und der Multimodalitätsdienst. Abbildung 1 verdeutlicht die Architektur des MMI-Systems graphisch. Die Evaluation von Gemeinsamkeiten der einzelnen Dienste zu einer gemeinsamen Dienstarchitektur, sowie die Integration der einzelnen Dienste zu einem Gesamtsystem sind ebenfalls Aufgabenbereiche des Projektes.

3 Beschreibung der TecO spezifischen Arbeiten

Das TecO beschäftigt sich im Rahmen dieses Projektes mit den Themen: Multimodalität, Penboard-spezifischen Diensten und Integration der einzelnen MMI-Dienste. Als ersten Einsatzbereich der Entwicklungen wird die institutsinterne Abwicklung des Reisekostenszenarios gewählt.

Multimodalitätsdienst

Innerhalb des TecOs stehen die Teledienste MMC und MMM zur Verfügung. Es erfolgt eine Auswahl von Funktionalitäten, die im Bereich der öffentlichen Verwaltung von Bedeutung sind. Für diese Funktionalitäten wird eine Benutzerschnittstelle erstellt, welche möglichst den Bedürfnissen der späteren Benutzer entspricht. Hier stellen sich nun Fragen wie: Was für Modalitäten stehen zur Verfügung? Wie können diese sinnvoll eingesetzt werden? Nicht alle Modalitäten sind gleichermaßen für bestimmte Aufgaben geeignet. Desweiteren darf nicht davon ausgegangen werden, daß alle in diesem Projekt untersuchten Modalitäten den Benutzern auch wirklich zur Verfügung stehen. Um einen freikonfigurierbaren Einsatz der Modalitäten erreichen zu können, wird zunächst untersucht, welche gemeinsamen Abstraktionen sich aus Benutzersicht für die unterschiedlichen Modalitäten definieren lassen. Hauptziele der Abstraktion von realen Modalitäten im Rahmen des Multimodalitätsdienstes sind somit: Geräteunabhängigkeit der Applikation und ein offener Ansatz, der das Einbinden zukünftiger Modalitäten erlaubt.

Zunächst werden die Modalitäten nach Eingabe- und Ausgabemodalitäten unterschieden. Zu den Eingabemodalitäten im konventionellen Sinn gehören textuelle Eingaben über Tastaturen und Eingaben über graphikorientierte Eingabegeräte wie z.B. Maus oder Stifteingaben. Ausgaben erfolgen zumeist über grafikfähige Dialogstationen. Bei genauerer Untersuchung der Gemeinsamkeiten einzelner Modalitäten lassen sich die folgenden Abstraktionen extrahieren: Texteingabe, Textausgabe, Graphikeingabe (Positioneingabe) und Graphikausgabe. Texteingabe(-ausgabe)modalitäten sind Geräte, die in der Lage sind eine Zeichenfolge des Grundalphabets zu liefern (auszugeben), z.B: Spracheingabe, Stifteingabe mit Handschrifterkennung (Touchscreen, Penboard, Sprachsynthese). Graphische Eingabegeräte, können als Quelle von Koordinatenpaaren angesehen werden. Meist gehören noch weitere Informationen, wie zum Beispiel der Zustand von Knöpfen zum Eingabetupel. Beispiele für Graphikein-/ausgabegeräte sind Penboard, Touchscreen und PDA. Für Daten, die von Rechnern nicht weiter interpretiert werden

können, oder wo eine solche Interpretation nicht erforderlich ist, können darüberhinaus durch die Quasi-Modalitäten Video und Audio vermittelt werden. Die gemeinsamen Abstraktionen ermöglichen nun die Nutzung von Modalitäten unabhängig von deren konkreter physikalischen Ausprägung.

Innerhalb dieses Projektes werden Schnittstellenfunktionen basierend auf einer zweistufigen Schnittstellenhierarchie angeboten. Auf der oberen Stufe der Hierarchie werden eigenständige Dienste angeboten, die das Bearbeiten kompletter Arbeitsgänge ermöglichen. Hierzu zählen die Dienstfunktionen *Konferenz*, *Besprechung*, *Dokument weitergeben*, *Eingang erhalten*. *Konferenz* basiert auf MMC und bietet die Möglichkeit des Einsatzes gemeinsamer Applikationen. *Besprechung* deckt den Bedarf an fernmündlicher Kommunikation ab. *Dokument weitergeben* und *Empfang erhalten* basieren auf MMM. Auf der darunterliegenden Ebene werden die feingranulareren Dienste angeboten. Es handelt sich dabei um das isolierte zur Verfügungstellen einzelner Komponenten, die auf der oberen Hierarchiestufe vereint angeboten werden.

Die Dienstleistungen auf der oberen Ebene bieten eine Benutzerschnittstelle an und können somit einerseits als eigenständige Teilanwendungen im Rahmen größerer Applikationen integriert werden, oder aber als komplett eigenständige Anwendungen eingesetzt werden. Je nach Dienstleistung muß bei Arbeitsbeginn noch nicht klar sein, welche Arbeitsgänge durchgeführt werden. Die feingranularen Dienste beschränken sich im Gegensatz hierzu auf eine reine Anwendungsprogrammierschnittstelle (API). Durch die selektive Auswahl einzelner Funktionalitäten ist die einfache Einbindung in verschiedene Kontexte vorstellbar, bei denen die Benutzerschnittstelle schon gegeben ist.

Penboard/PDA spezifische Dienste

Der Grundgedanke beim Einsatz eines Penboards, ist die Auflösung von Ein-Personen-Arbeitsplätzen für interpersonelle Arbeitsvorgänge mehrerer Personen am gleichen Ort. Hier bietet sich vor allem der Vorteil der großen Darstellung, die auch noch ein Erkennen aus größerer Entfernung möglich macht. Zur Eingabe muß der Anwender allerdings trotzdem noch vor dem Penboard stehen. Abhilfe kann hier, in eingeschränktem Maße das *Telepointing* bieten. Mithilfe eines PDA's kann das Zeigen auf dem Penboard ermöglicht werden, sodaß ein Benutzer auch von größerer Entfernung, z.B. seinem Platz innerhalb einer Konferenzrunde auf Objekte zeigen kann um seine Diskussionskriterien zu verdeutlichen.

Ein weiterer Einsatz des Penboards, neben dem innerhalb einer Konferenz, bildet das Pinwandszenario. Die Veröffentlichung von Information erfolgt herkömmlich häufig über das Anstecken an einem Anschlagbrett. Wird das Anschlagbrett durch ein Penboard ersetzt, so können Aktionen wie Anbringen oder Abhängen von Information, neben der Abrufmöglichkeit der eigentlichen Informationen zu jedem Zeitpunkt, komfortabel vom Arbeitsplatz aus erfolgen. Diese Arbeitserleichterung macht sich vor allem bemerkbar, wenn mehrere Anschlagbretter, an unterschiedlichen Lokationen verwaltet werden müssen. Die Veröffentlichung kann von zentraler Stelle aus erfolgen, wobei Informationen in Form von Texten oder Bildern fachbereichsgemäß auf dem Penboard angeordnet werden können.

Integrationskonzept

Im ersten Prototyp des Multimodalitätsdienstes werden ausschließlich Modalitäten zum Einsatz kommen, die im TecO verfügbar sind. Prinzipiell kann die Benutzung der MMTS Dienste, aufgrund der Abstraktion von Modalitäten, auch über Modalitäten des Gesamtprojektes, wie Spracheingabe oder Stift/Touchscreen erfolgen. Die Integration der einzelnen physikalischen Dienste kann durch die Zuordnung zu der entsprechenden Modalitätenklasse, sowie dem Wissen über beim Anwender tatsächlich verfügbare physikalische Modalitäten erfolgen. Die so ermöglichte benutzerspezifisch flexible Konfigurationsfunktionalität kann ebenfalls in einen eigenen Dienst ausgelagert werden. Durch die Unterstützung von Flexibilität besteht auch indirekt ein Bezug zur Adaptivität, beispielsweise wenn beim Anwender Behinderungen hinsichtlich der Nutzung einer Modalität bestehen.

Im Zuge der Verallgemeinerung des MMI-Projektes in Richtung generischer Schnittstellen, können in den Multimodalitätsdienst über die Dienstschnittstelle DCE weitere Anwendungen integriert werden. Die Modalitätenzuordnung für beliebige Anwendungen bleibt zum gegenwärtigen Zeitpunkt allerdings noch ein Fernziel.

Anwendungsszenario

Die TecO spezifischen Dienste werden mithilfe der Realisierung des Reisekostenszenarios validiert. Reisekostenabrechnung ist eine Anwendung, die in jeder öffentlichen Verwaltung benötigt wird.

Die Durchführung des Szenarios erfolgt folgendermaßen: Zuerst findet eine Gruppenbesprechung über eine eventuelle Dienstreise der Mitarbeiter im TecO und Herrn Schneider im Rechenzentrum statt. Hierfür bietet sich der Dienst *Besprechung* an. Dann füllt der Reisende elektronisch den Dienstreiseantrag aus, und schickt ihn mit Hilfe des Dienstes *Dokument weitergeben* an Herrn Schneider. Herr Schneider empfängt diesen Antrag mithilfe des Dienstes *Eingang erhalten*. Bei Unstimmigkeiten erfolgt nun eine Rücksprache mit den Mitarbeitern. Hierzu dient der Dienst *Konferenz*, bei welchem der Dienstreiseantrag als Diskussionsgrundlage eingebracht werden kann.

Es kommen alle Dienste des Multimodalitätsdienstes auf hoher Ebene, und die Telekonferenz mithilfe eines Penboards zum Einsatz. Da die Dienste auf hoher Ebene intern auf die feingranularen Dienste aufsetzen, werden diese automatisch mit abgehandelt.

4 Durchführung des Verbundprojektes

Einführend erfolgte zunächst eine Analyse, was für Anwendungen und Szenarien in den Bereichen Vorgangsbearbeitung und Bürokommunikation in öffentlichen Verwaltungen gängig sind. Bei der Analyse wurden MMI-relevanten Aspekten und der Möglichkeit des Einsatzes von Telekooperation besondere Beachtung beigemessen.

Aufbauend auf den Ergebnissen der Analyse wurde ein Konzept erstellt zur Unterstützung und Verbesserung der Vorgänge und Nutzung der Telekooperation. Hierbei wurden Techniken und Verfahren in den Bereichen Multimodalität, Sprachverarbeitung, taktile MMI-Verfahren, wie Stifteingabe, Touchscreen, und adaptive MMI-Agents miteinbezogen.

Gegenwärtig befindet sich das Projekt in der Phase der Implementierung. Der Realisierung folgt eine Pilotierung der Prototypen in realen Anwendungsszenarien. Zunächst kommen die einzelnen Systemkomponenten Multimodalität, Sprachverarbeitung, taktile und adaptive MMI-Agents in Anwendungsszenarien, welche unabhängig voneinander sind, zum Einsatz. So können die jeweiligen Szenarien individuell stark auf die Zielbereiche der jeweiligen Systemkomponenten zugeschnitten werden.

Weiterführend wird ein Integrationskonzept für die Realisierung einer intelligenten Mensch-Maschine-Schnittstelle entwickelt werden, bei der die einzelnen Komponenten gemeinsam berücksichtigt werden. Dieses Gesamtsystem wird dann ebenfalls anhand eines Anwendungsszenarios demonstriert werden.

Parallel zu diesen Entwicklungsarbeiten wurde ein Katalog von arbeits- und kognitionspsychologischen Kriterien für den Entwurf und die Evaluierung der einzelnen Systemkomponenten und des Gesamtsystems erstellt und wird nun eingesetzt. Ebenfalls parallel wird gegenwärtig ein Qualitätssicherungs- und Optimierungskonzept für die Teilkomponenten des MMI-Systems entwickelt und genutzt. Abschließend werden Richtlinien und Kriterien zur Einführung der Systeme nach technischen, organisatorischen und wirtschaftlichen Kriterien erstellt werden.

5 Organisation des Verbundprojektes

Auftraggeber des Verbundprojektes ist die **DeTeBerkom**. Das Projekt umfaßt den Arbeitszeitraum vom **1.6.1994 bis zum 31.5.1996**. Es wird mit folgenden Projektpartnern als Verbundprojekt durchgeführt:

- Das TUBKOM DAI-Labor, TU Berlin (**TUB-DAI**) übernimmt die Verbundprojektkoordination. Sie befaßt sich mit den Arbeitsbereichen *Adaptivität, Dienstarchitektur* und *Integration*.
- Der Interdisziplinäre Forschungsverbund "Technikvorsorge und Folgenforschung", TU Berlin (**TUB-TVFF**) erstellt *arbeits- und kognitionspsychologische Kriterien*.
- Die Unternehmensberatung für integrierte Systeme GmbH, Berlin (**UBIS**) erstellt ein *Qualitätssicherungs- und Optimierungskonzept* und übernimmt die *grundsätzliche Einführungsstrategie des MMI-Systems*.
- Das Telecooperation Office, TU Karlsruhe (**TecO**, Stefan Gessler, Sonja Zwißler) arbeitet im Bereich der *Multimodalität, an Penboard-spezifischen Diensten*, und am *Integrationskonzept*.
- Die IBM GmbH, European Networking Center (**IBM**) stellt einen *Sprachverarbeitungsdienst* zur Verfügung.
- Das Fraunhofer-Institut für Arbeitswissenschaft und Organisation, Stuttgart (**FhG-IAO**) widmet sich dem Thema *Stifteingabe / Touchscreen*.
- Das Fraunhofer-Institut für Graphische Datenverarbeitung, Darmstadt (**FhG-IGD**) arbeitet im Bereich der *Flexibilität und Transparenz*.

Der BERKOM Multimedia Mail Teledienst (MMM)

Stefan Gessler

1 Projektrahmen

Im Rahmen des Projektes BERKOM II der DeTeBerkom GmbH, einer Tochter der Deutschen Telekom, wurde mit *Multimedia Mail* ein Teledienst entwickelt, der die asynchrone kooperative Zusammenarbeit verschiedener Personen auf elektronischem Wege ermöglicht bzw. unterstützt.

Zentraler Aspekt des Projektes war die Unterstützung des Teledienstes durch die meisten der heute üblichen Rechnerplattformen auf der Basis von ISO- und CCITT- Standards bzw. Empfehlungen. Das Telecooperation Office war an der Entwicklung dieses Teledienstes im Auftrag von Digital CEC Karlsruhe beteiligt. Im Dezember 1994 konnte die erste Projektphase sehr erfolgreich abgeschlossen werden, eine zweite Phase ist derzeit in Planung.

In diesem Abschnitt wird der BERKOM Teledienst Multimedia Mail vorgestellt. Der Abschnitt ist nach folgenden Gesichtspunkten unterteilt:

- Inhalt von MMM
- Erweiterung um externe Referenzen
- Erweiterung um Hypermedia Funktionalität
- Architektur

2 Projektbeschreibung

Der BERKOM Multimedia Mail Teledienst [MSS93] stellt einen Dienst zum Austausch multimedialer Nachrichten basierend auf dem Message Handling System (MHS) X.400 zur Verfügung. Interworking mit anderen Datenaustauschmechanismen ist u.a. für MIME und für den Telefaxdienst vorgesehen.

Eine Nachricht enthält Header-Informationen (z.B. Sender, Empfänger etc.) und Nachrichtendaten, sog. Bodyparts. Im Gegensatz zum bekannten textbasierten Mailedienst kann BERKOM-MMM beliebig viele Bodyparts in unterschiedlichen Datenformaten enthalten (siehe auch Abb. 45). Diese Datenformate können folgende Ausprägungen annehmen:

1. Text
2. Graphik und Festbild
3. Bewegtbild (Video)
4. Audio
5. Audio/Video
6. Dokumentenformat Postscript
7. Binary-Formate

Für BERKOM-MMM wurden Anforderungen definiert, die durch nichtstandardisierte Erweiterungen realisiert wurden. Im einzelnen handelt es sich hierbei um *externe Referenzen* und um eine *Hypermediafähigkeit*. Diese Erweiterungen werden im folgenden vorgestellt.

3 Erweiterung um externe Referenzen

Durch den wesentlich höheren Speicherbedarf multimedialer Nachrichten wurde neben dem X.400 MHS ein weiterer Austauschmechanismus in den Dienst integriert. Dieser Austauschmechanismus stellt eine Realisierung des *Distinguished Object Reference (DOR)* Konzeptes in Verbindung eines sog. globalen Speichers dar.

Neben dem aufwendigen Versenden nach dem *Store-and-Forward-Prinzip*, hat der Originator einer Nachricht die Möglichkeit, den großvolumigen Nachrichtenteil über Kommunikationskanäle mit hoher Bandbreite in einem sog. globalen Speicher abzulegen. Dieser globale Speicher liefert im Gegenzug eine Referenz auf die Daten zurück (DOR). Diese Referenz wird nun in die Nachricht integriert und dem Empfänger ausgeliefert. Der Empfänger kann mithilfe dieser Referenz die eigentlichen Daten ebenfalls wieder über breitbandige Kommunikationskanäle abrufen. Abbildung 45 veranschaulicht dieses Konzept.

Um eine Referenz (DOR) zu versenden, steht als weiterer Datentyp die

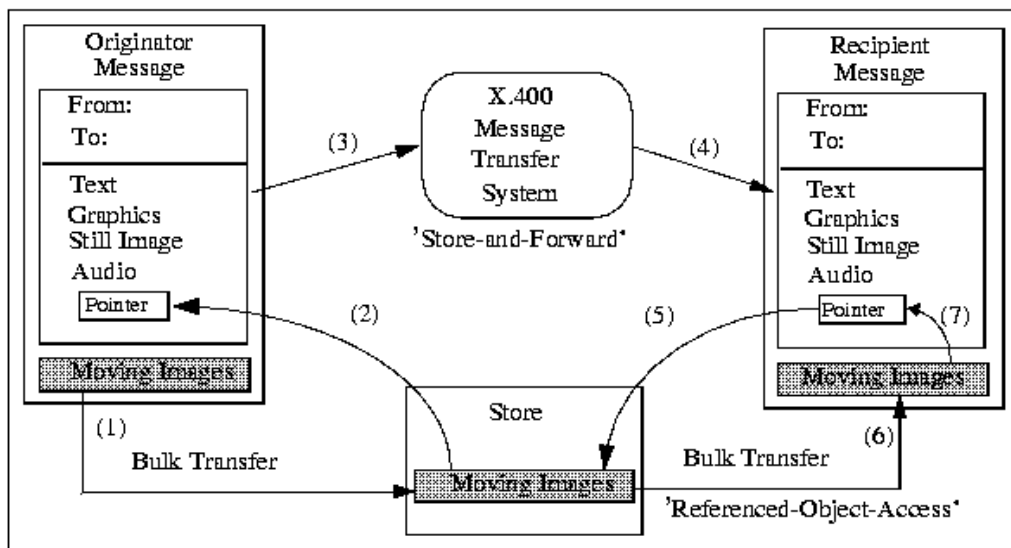


Abbildung 45. Store-and-Forward Prinzip und DOR-Mechanismus

- externe Referenz (Xref)

zur Verfügung. Der globale Speicher-Mechanismus ist als realisiert und erlaubt eine Vielzahl von Anwendungsvariationen. Das Ablegen wie auch das Abrufen von Dateien ist ein hochinteraktiver Vorgang.

4 Erweiterung um Hypermedia-Funktionalität

Typischerweise stehen verschiedene Teile einer Nachricht in einer gewissen Relation zueinander. Um diese Relationen dem Empfänger der Nachricht vermitteln zu können, stellt BERKOM-MMM einen Hypermedia-Mechanismus zur Verfügung, um solche Zusammenhänge (sog *Links*) beschreiben zu können.

Während Hypermediamechanismen für eindimensionale (textuelle) Daten seit Jahren bekannt sind, müssen ähnliche Konzepte für mehrdimensionale, insbesondere auch für zeitkontinuierliche Daten im Rahmen dieses Projektes definiert werden. Hier spielen bereits die Handhabbarkeit und das intuitiver Verständnis der Benutzer eine große Rolle. Das Beschreiben von Relationen stellt hohe Anforderungen an die Mensch-Maschinen-Interaktion.

Analog zu externen Referenzen wurde auch für Hypermediainformationen ein weiterer Nachrichtentyp eingeführt:

- der Linkteil

(die Informationen über alle Links innerhalb einer Nachricht werden in *einem* Nachrichtenteil abgelegt, sodaß eine Nachricht nur *eine* Linkdatei enthalten kann)

5 Architektur

Die Architektur des BERKOM Multimedia Mail Teledienstes ist in Graphik 46 dargestellt.

Man erkennt die drei Hauptkomponenten

- User Agent
- Global-Store-Server
- X400 MHS

Ebenfalls aufgeführt sind hier die Interoperabilitätskomponenten zu weiteren, externen Kommunikationsdiensten (Telefax, etc.), die nicht Bestandteil der allgemeinen Spezifikation sind.

Abbildung 46. MMM-Architektur

6 Zusammenfassung

Der BERKOM-Teledienst MMM wurde als Multi-Vendor-Projekt von verschiedenen Firmen und Forschungseinrichtungen für jeweils eine bestimmte Rechnerplattform entwickelt. Lediglich der Xref-Dienst, der Global-Store und die Interoperabilitätskomponenten wurden von nur einem Projektpartner realisiert.

Das Telecooperation Office war an der Entwicklung des Dienstes für die Plattform Digital-Alpha-AXP unter OSF1 beteiligt. Entwicklungspakete waren dabei die Erzeugung und Verwaltung der Externen Referenzen, die Kommunikation mit dem Global Store, z.T. auch der die graphische Benutzerschnittstelle und User Agent Funktionalitäten. Außerdem wurde am TecO die projektexklusive Interoperabilitätskomponente zum Telefaxdienst spezifiziert und realisiert.

Die Arbeiten konnten im Rahmen der Internationalen Funkausstellung in Berlin 1993 und der CeBit in Hannover 1994 erfolgreich präsentiert werden.

Das BerKom-II-Projekt MMT

Markus Hofmann, Claudia Schmidt

1 Überblick

Im Rahmen des *Berliner-Kommunikationssystems-II* (BERKOM-II) entsteht mit dem *Multimedia-Transportsystem* (MMT) [MMT94] eine Kommunikationsplattform für verteilte Multimedia-Anwendungen. Parallel zu diesem Projekt werden im Anwendungsbereich die beiden Teleservice-Projekte *Multimedia-Collaboration-Service* (MMC) und *Multimedia-Mail-Service* (MMM) durchgeführt, die Dienste zur Unterstützung von Audio-Video-Konferenzen und zum Austausch von Multimedia-Dokumenten zur Verfügung stellen. Diese beiden Anwendungen nutzen den MMT-Protokollturm als Kommunikationssystem. Die Infrastruktur besteht derzeit aus LANs (FDDI, Ethernet, Token Ring), die über WANs miteinander verbunden sind. In der aktuellen Projektphase erfolgt der Übergang auf ATM-basierte Netzwerke

In das MMT-Projekt involviert sind die Firmen DEC (vertreten durch das CEC, Campus-based Engineering Center), IBM (vertreten durch das ENC, European Networking Center) und Siemens/SNI sowie die Forschungseinrichtungen Gesellschaft für Mathematik und Datenverarbeitung (GMD) und die Technische Universität Berlin. Das Institut für Telematik unterstützt DEC als Kooperationspartner.

Das BERKOM-Multimedia-Transportsystem zeichnet sich vor allem durch die Unterstützung verschiedener Dienste sowie durch die Integration von Multicast aus. Der Einsatz des *Internet-Stream-Protocol-II* (ST-II) [Top90] in der Vermittlungsschicht und dem *Xpress-Transfer-Protocol-Lite* (XTP-Lite) als Transportprotokoll ermöglicht die Gewährleistung zuvor ausgehandelter Dienstgütern. Die Architektur des Transportsystems ist in Abbildung 47 dargestellt.

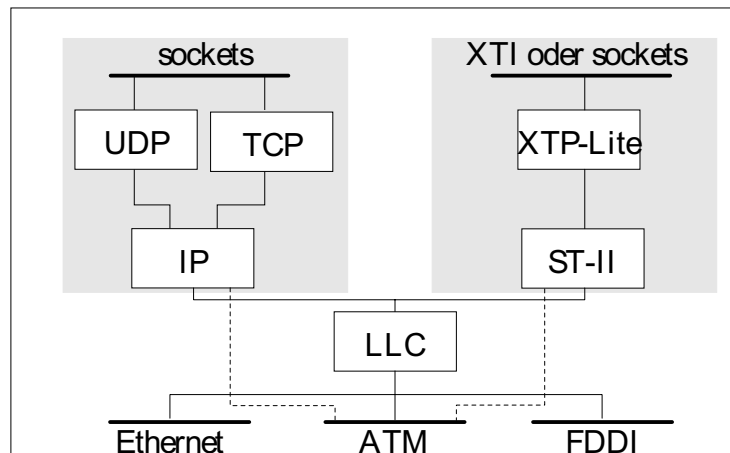


Abbildung 47. Architektur des BERKOM-II-MultiMedia-Transportsystems

Vom Institut für Telematik der Universität Karlsruhe wurde in Zusammenarbeit mit weiteren Projektpartnern das Transportprotokoll XTP-Lite spezifiziert und anschließend implementiert.

2 ST-II - das Vermittlungsprotokoll des MMT

Für die Vermittlungsschicht wurde von den MMT-Partnern das Internet-Stream-Protocol, Version 2 (ST-II) gewählt, da dieses Protokoll einige Charakteristiken (Multicastfähigkeit, Ressourcenreservierung) besitzt, die von Multimedia-Anwendungen benötigt werden. Zum Zeitpunkt der Spezifikation des BERKOM-Protokollturms existierten keine alternativen Protokolle mit ähnlichen Fähigkeiten. Momentan wird in der Projektgruppe ein Ersatz von ST-II durch das flexiblere RSVP diskutiert [SZ95d]. Gleichzeitig wird aber auch ST-II in einer Arbeitsgruppe der IETF (Internet Engineering Task Force) weiterentwickelt.

ST-II bietet einen verbindungsorientierten, aber unzuverlässigen Dienst. Die Kommunikationsstruktur, auf der ST-II basiert, ist ein sogenannter *Flow*, ein gerichteter Baum vom Sender über alle Zwischensysteme zu der Empfängergruppe. In jedem Kommunikationssystem ist eine ST-II-Instanz, ein sogenannter *ST-Agent* lokalisiert. ST-II unterstützt Multicast durch den *ST-II-Flow*, eine unidirektionale 1:n Verbindung. Diese kann zu einer m:n Verbindung erweitert werden, indem von jedem Mitglied ein Multicastbaum aufgebaut wird. Weiterhin können einzelne ST-II-Verbindungen als Gruppen organisiert und durch das Gruppenmanagement von ST-II verwaltet werden.

Funktionen zum Ressourcenmanagement spezifiziert ST-II nicht explizit, aber in der Protokollspezifikation wird durch die sogenannte *FlowSpec* eine Datenstruktur für die Aushandlung der Dienstqualität vorgeschlagen. Diese *FlowSpec* der ST-II-Spezifikation wurde für BERKOM modifiziert, so daß sie die im BERKOM Kontext relevanten Qualitätsparameter widerspiegelt. Das ST-II-Protokoll trifft eine strikte Trennung zwischen dem Transfer der Daten und Kontrollfunktionen. Alle Funktionen, die der Verwaltung des Datentransfers dienen (Verbindungsaufbau, Modifikation der Verbindungscharakteristik, Fehlerbehandlung), behandelt das in ST-II integrierte Protokoll *ST Control Message Protocol (SCMP)*. Den Datentransfer selbst unterstützt ST-II durch die Verwendung von kurzen Paketköpfen, womit eine schnelle Bearbeitung der Pakete ermöglicht wird.

3 XTP-Lite - das Transportprotokoll des MMT

Das XTP-Derivat XTP-Lite liegt im Protokollturm oberhalb des verbindungsorientierten ST-II-Protokolls. Die aktuelle XTP-Lite-Spezifikation unterstützt zuverlässige und unzuverlässige, bidirektionale Punkt-zu-Punkt-Verbindungen sowie unidirektionale, unzuverlässige Multicast-Verbindungen. Dazu wird der Multicast-Dienst von ST-II genutzt und um Managementfunktionen erweitert. Die Integration eines zuverlässigen Multicast-Dienstes ist in Planung. Zur Dienstgüteunterstützung bildet XTP-Lite die Qualitätsparameter der Benutzerschnittstelle auf diejenigen Parameter ab, die ST-II zur Aushandlung der Dienstqualität benötigt. Durch die Spezifikation von Qualitätsparametern ist der Anwender in der Lage, speziell auf seine Anforderungen zugeschnittene Dienste anzufordern, z.B. einen zuverlässigen Dienst oder einen Dienst für die Übertragung isochroner Datenströme.

Die Protokollfunktionen von XTP-Lite sind ebenfalls auf die sehr unterschiedlichen Einsatzgebiete abgestimmt. Die Verbindung kann alternativ durch einen 2-Wege-Handshake oder implizit aufgebaut werden. Bei einem impliziten Aufbau wird der Wunsch zum Verbindungsaufbau mit dem ersten Datenpaket angezeigt. Die verschiedenen Mechanismen zur Fehlererkennung und -behandlung sind abhängig von der gewählten Dienstqualität aktivierbar und schließlich werden spezielle Protokollmechanismen wie z.B. selektive Übertragungswiederholung eingesetzt, die eine effiziente Übertragung ermöglichen.

Der Übergang auf ATM-basierte Netzwerktechnologie in der aktuellen Projektphase ermöglicht die Reservierung von Netzressourcen. Zusammen mit entsprechender Betriebssystemunterstützung kann somit eine ausgehandelte Dienstgüte für die Dauer der Kommunikation garantiert werden.

4 Die MMT-Dienstschnittstelle

Die Anwendungsschnittstelle von MMT wurde im Hinblick auf eine Nutzung durch moderne Applikationen konzipiert. Insbesondere Multimedia-Anwendungen verlangen sehr unterschiedliche Dienste und damit einen erweiterten Satz von Parametern und eine größere Flexibilität bei der Auswahl dieser Parameter an der Schnittstelle zum Transportsystem. MMT bietet diese Flexibilität durch eine Menge von Qualitätsparametern und die Auswahl zwischen Dienstklassen, die beschreiben wie die ausgehandelten Garantien zu interpretieren sind.

Der MMT-Transportdienstbenutzer hat die Möglichkeit, die von ihm benötigte Dienstqualität detailliert an der Schnittstelle zum Transportsystem zu spezifizieren. Die Parameter werden von XTP-Lite auf ST-II-Parameter abgebildet und anschließend handeln die ST-II-Instanzen die Dienstqualität der Verbindung aus. Dabei ist es möglich eine verschiedene Dienstqualität für die beiden Richtungen einer Duplexverbindung zu wählen. Das MMT-Transportsystem stellt folgende Qualitätsparameter zur Verfügung:

- Maximale TSDU-Größe
- Durchsatz/Rate
- Ende-zu-Ende Verzögerung
- Zuverlässigkeit

Die ersten drei Parameter werden in MMT jeweils durch einen gewünschten und einen Grenzwert beschrieben. Eine Verbindung wird nur dann aufgebaut, wenn die Grenzwerte eingehalten werden können. Der gewünschte Wert wird während des Verbindungsaufbaus auf den real erbringbaren Wert reduziert, solange er den Grenzwert nicht erreicht. Nach erfolgreichem Verbindungsaufbau wird dem Transportdienstbenutzer die resultierende Dienstqualität angezeigt.

Für die Zuverlässigkeit wurden fünf verschiedene Typen definiert, deren Spektrum von einem unzuverlässigen Dienst mit oder ohne Fehleranzeige bis zu einem zuverlässigen Dienst reicht. Im einzelnen existieren die Typen:

- Typ 0: Ignorieren von korrupten und verlorenen TSDUs
- Typ 1: Ignorieren von korrupten TSDUs, Anzeigen von verlorenen TSDUs
- Typ 2: Anzeigen von korrupten und verlorenen TSDUs
- Typ 3: Ignorieren von korrupten TSDUs, korrigieren von verlorenen TSDUs
- Typ 4: Korrigieren von korrupten und verlorenen TSDUs

Das Multimedia-Transportsystem garantiert dem Anwender die ausgehandelte Dienstqualität für die Dauer einer Verbindung. Die Garantie ist dabei abhängig von einer sogenannten Dienstklasse, die beim Verbindungsaufbau gewählt wird. In der *best-effort*-Klasse muß die Dienstqualität lediglich für ein mittleres Verkehrsaufkommen eingehalten werden, im Gegensatz dazu garantiert das Transportsystem für die *garantierte*-Klasse, daß die Dienstqualität auch bei stoßartigem Verkehr nicht verletzt wird, d.h. wenn Anwendungen längerfristig mit einer sehr hohen Rate senden, ist es möglich, daß die Dienstgarantien der *best-effort*-Klasse nicht mehr eingehalten werden.

5 Implementierung

Die einzelnen BERKOM-Partner implementieren das Transportsystem auf verschiedenen Plattformen (IBM, DEC, HP, SUN, Siemens), wobei das Institut für Telematik in Kooperation mit DEC eine ALPHA-Workstation unter dem Betriebssystem OSF/1 als Implementierungsplattform verwendet. Das Transportsystem ist vom Anwender über zwei verschiedene Schnittstellen (X/OPEN Transport Interface (XTI) und BSD Socket Interface) ansprechbar, von denen jeweils eine von den Partnern unterstützt werden muß. Die Implementierung des Transportsystems am Institut für Telematik wird über eine erweiterte BSD-Socket-Schnittstelle angesprochen. Erweiterungen der Schnittstelle wurden bei der Realisierung des Multicast-Dienstes notwendig, da einige der benötigten Dienstprimitive nicht ohne weiteres auf die definierten BSD-Socket-Operationen umzusetzen sind.

Die aktuelle Implementierung unterstützt alle verpflichtenden Teile der MMT-Spezifikation, Version 5 [MMT94]. Dazu zählen Verbindungsmanagement und Datentransfer für Punkt-zu-Punkt- und Multicast-Verbindungen. Für die letzteren wird weiterhin ein dynamisches Gruppenmanagement durch das Hinzufügen und Entfernen von Empfängern unterstützt. Die Implementierung besitzt Funktionen zur Spezifikation von Qualitätsparametern, zum Abbilden und Aushandeln sowie Neuaushandeln der Dienstqualität. Zusätzlich wurde optional eine ratenbasierte Flußkontrolle implementiert und Erweiterungen zur Dienstqualität spezifiziert. Zum Testen und Demonstrieren der Implementierung wurden außerdem mehrere graphische Testanwendungen erstellt. Die Implementierung des Transportsystems und der Anwendungen wird im Rahmen einer ständigen Demonstration bei der DeTeBerkom in Berlin vorgeführt.

6 Aktuelle Projektarbeiten

In der laufenden Projektphase erfolgt die Integration eines zuverlässigen Multicast-Dienstes in das Transportsystem. Hierzu sind Konzepte zur Gruppenverwaltung und effiziente Mechanismen zur Fehlerbehebung zu entwickeln, zu spezifizieren und zu implementieren. Zugleich erfolgt der Entwurf und die Implementierung einer Monitor-Funktion zur Überwachung der unterstützten Qualitätsparameter. Der Anwender hat hierbei die Möglichkeit, zu Testzwecken die aktuellen Werte der einzelnen Parameter über eine graphische Anwendung abzufragen und sich diese in graphischer Form darstellen zu lassen.

Darüberhinaus ist der Übergang auf ATM-basierte Netzwerke sowie die Kopplung über ISDN vorgesehen, wofür bereits zahlreiche Konzepte in den Bereichen Internetworking, Adressierung, Dienstgüteunterstützung, Protokollarchitektur und Netzwerkmanagement erarbeitet wurden. Die Arbeiten sind im einzelnen den folgenden Bereichen zuzuordnen und werden von den Mitarbeitern Stefan Dresler, Markus Hofmann, Claudia Schmidt und Hajo Wiltfang bearbeitet:

- *Dienstgüteunterstützung*: Definition einer Abbildung der MMT-FlowSpec auf Quality-of-Service Parameter von ATM, Darstellung des Einflusses der LAN-Emulation auf die Dienstunterstützung innerhalb des MMT, Untersuchung des Einflusses von Multicast-Kommunikation auf Ressourcenreservierung in Zwischen- und Endsystemen.
- *Internetworking*:
Definition von Adressierungsformaten unter Berücksichtigung von Adreßauflösungskonzepten für WAN-Umgebungen, Untersuchung von Realisierungsmöglichkeiten von ARP-Servern, Erarbeitung eines Gesamtkonzept für das Internetworking aller BERKOM-Partner.
- *Netzwerkmanagement*: Erarbeitung von Vorschlägen zur Erweiterung der Management Information Base (MIB) für ATM.
- *MMT über ISDN*: Untersuchung verschiedener Anwendungsschnittstellen, Abbildung von Adressierungsformaten, Internetworking.

Neben den genannten Punkten werden derzeit Konzepte zur Realisierung von Multicast in ATM-Umgebungen und die Auswirkungen von Multicast auf die Belastung von ATM-basierten Netzen untersucht. Daneben sind Signalisierungskonzepte im Hinblick auf ihre Eignung für Multicast auszuwählen und zu modifizieren. Im Bereich des Traffic Shaping sind Vorschläge für heterogene Switch-Umgebungen zu erarbeiten. Als Ergänzung zu den MMT-spezifischen Arbeiten wird eine Studie über zukünftige Internet-Protokolle angefertigt, die neuartige Ansätze diskutiert und mögliche Trends innerhalb der Internet-Gemeinde aufzeigt.

Multimediale Online-Dienste nach dem kombinierten WWW-Ansatz

Stefan Gessler, Robert Wicke

1 Einleitung

Das *World Wide Web* (WWW) ist ein Informationssystem, das es ermöglicht, eine Reihe der im Internet angebotenen Informationsdienste (FTP, gopher, News, E-Mail) und Dokumente unterschiedlichster Formate über eine konsistente und leicht zu bedienende Benutzerschnittstelle in Form eines Hypertextsystems zugänglich zu machen. Ursprünglich 1989 vom CERN in Genf für den Informationsaustausch zwischen räumlich verteilten Forscherteams gedacht, hat das WWW inzwischen gerade wegen seiner einfachen Bedienbarkeit eine Popularität weit über den akademischen Bereich hinaus erreicht, und spätestens seit 1994 kann bei seiner Ausbreitung von einem wahren Boom gesprochen werden. Immer mehr Firmen entdecken das WWW inzwischen auch als ein geeignetes Medium für das Angebot kommerzieller Dienste (z.B. Kundensupport, Online-Bestellungen), für die 1995 der Durchbruch erwartet wird.

Das Telecooperation Office führt im Auftrag der SAP AG eine Studie durch, im Rahmen derer verschiedene Möglichkeiten der Nutzung des *World Wide Web* (WWW) für kommerzielle Online-Dienste untersucht werden. Ein Schwerpunkt liegt dabei in der Integrierbarkeit unterschiedlichster Medientypen innerhalb eines WWW-Dienstes. Diese Ergebnisse fließen in die Entwicklung eines Prototypen für einen Online-Dienst der SAP AG ein..

2 Überblick

Neben dem Schwerpunkt der Medienintegration soll die Studie einen Überblick über existierende Online-Dienste geben und Tools zur Unterstützung ihrer Entwicklung sowie Software zu ihrem Betrieb vorstellen.

Die Studie gliedert sich in folgende Punkte:

- Existierende Online-Dienste
- Software für Entwicklung und Betrieb von Online-Diensten
- Medienintegration
- Prototyp

2.1 Existierende Online-Dienste

Es werden Beispiele sowohl für gelungene als auch für weniger gelungene kommerzielle Online-Dienste im WWW vorgestellt und Art und Umfang der darin angebotenen Dienstleistungen aus den Bereichen Marketing, Service und Support und Ordering Services betrachtet. Ein weiterer Schwerpunkt sind gegenwärtige und zukünftige Möglichkeiten zur Durchführung sicherer Geschäftstransaktionen über das Internet.

2.2 Software für Entwicklung und Betrieb von Online-Diensten

Neben Filtern zur Konvertierung bestehender Dokumente in das WWW-eigene HTML-Format wird vor allem ein Überblick über Authoring-Systeme gegeben, die zur Erstellung größerer WWW-Dienste unerlässlich sind.

Ferner werden neben der für das Angebot eines WWW-Dienstes benötigten Server-Software die z.Z. erhältlichen Browser besprochen, die das Navigieren im WWW und das Betrachten von HTML-Seiten ermöglichen.

2.3 Medienintegration

Als Schwerpunkt der Studie wird die Einbindung verschiedener Medientypen (Audio, Video, verschiedene Text- und Hypertextformate) in WWW zum Angebot eines multimedialen Online-Dienstes betrachtet. Weitere Punkte sind die Einbindung großer Datenbestände, die lokal (z.B. auf CD-ROM) vorliegen, sowie eine Verbindung der asynchronen Interaktion über WWW mit synchronen Telekonferenzszenarien und Application Sharing.

2.4 Prototyp

Abschließend ist ein generischer Prototyp eines kommerziellen Online-Dienstes zu erstellen, an dem die gewonnenen Erkenntnisse über Medienintegration etc. demonstriert werden und der Eindruck von einem möglichen durch SAP in Zukunft angebotenen Dienst im WWW vermitteln soll.

2.5 Gegenwärtiger Status

Obwohl die Zusammenarbeit mit SAP erst vor wenigen Wochen begann, konnte bereits während einer Präsentation erster Ergebnisse ein wertvoller Beitrag im Rahmen einer Strategieberesprechung auf höchster Ebene innerhalb von SAP geleistet werden.

Der Abschluß der ersten Phase ist für den Juni '95 geplant. Eine anschließende zweite Phase bis Dezember '95 ist bereits im Gespräch.

SFB 346: Rechnerintegrierte Produktion und Fertigung von Bauteilen

Arnd G. Grosse, Ludwig Keller, Dietmar A. Kottmann

1 Einleitung

Moderne ingenieurwissenschaftliche Organisationsstrukturen, wie sie unter den Schlagwörtern „fraktale Fabrik“ und „Simultaneous Engineering“ diskutiert werden, zeichnen sich in zunehmenden Maße durch eine hohe informationstechnische Vernetzung der einzelnen durch EDV-Werkzeuge gestützten Aktivitäten aus. Sie erfordern zum einen einen flexiblen Datenaustausch zwischen den verschiedenen Anwendungssystemen und zum anderen die Möglichkeit der persistenten Datensicherung.

Ihre Realisierung setzt eine integrierte Informationstechnik voraus, die die Kooperation von verschiedenen Anwendungen und das Ausnutzen von verfügbarem Wissen möglichst effizient unterstützen. Technische Anforderungen an die Informationstechnik wurden bisher in erster Linie im Kontext der Vereinheitlichung der Datenverwaltung (STEP) vollzogen. Anforderungen hinsichtlich der informationstechnischen Vernetzung wurden hingegen nur wenig untersucht (Ansätze für eingeschränkte Teilbereiche sind z. B. MAP, VDA).

Im Rahmen der Arbeiten des Instituts für Telematik im SFB 346 wurde eine empirische Studie zur Analyse der Anforderungen von Maschinenbauanwendern an Verteilungsmechanismen durchgeführt und die am Institut entwickelten technischen Konzepte an diesen Anforderungen gespiegelt. Diese Spiegelung wird im Rahmen des folgenden Beitrags in ihren wichtigsten Aspekten wiedergegeben. Zugleich wird auf aktuelle Erweiterungen eingegangen, die auf den Erkenntnissen der Studie und auf den Arbeiten an aktuellen Prototypen beruhen. Die hier zu findenden Arbeiten werden durch die Analyse von Fehler-toleranzfragen im System Agora (Beitrag S. 39), die Untersuchung der Datenhaltung im Kontext des Mobile Computing (Beitrag S. 43) und der Analyse moderner Datenbankarchitekturen am Beispiel von ObjectStore (Beitrag S. 27) ergänzt.

2 Zentrale Ergebnisse der empirischen Studie

Im Rahmen der empirischen Studie, die unter Beteiligung von 23 Maschinenbauunternehmen durchgeführt wurde, haben sich die folgenden zentralen Ergebnisse herauskristallisiert:

1. Die gemeinsame Datennutzung wird stark zunehmen. Der Datenaustausch erfolgt dabei bidirektional, wird sich aber im wesentlichen auf Kommunikation innerhalb eines Unternehmens beschränken.
2. Die Schäden aufgrund von Nichtverfügbarkeit von Daten werden zukünftig signifikant steigen.
3. Die zunehmende Integration der Unternehmensbereiche Konstruktion, Planung, Fertigung und Forschung/Entwicklung erfordern die nichtreplizierte Datenhaltung mit hoher Aktualität.
4. Externe Unternehmensbereiche, wie z.B. Vertrieb und Zulieferer, erfordern hingegen aus Autonomiegründen die Mehrfachhaltung der Daten unter abgeschwächten Konsistenzbedingungen.
5. Der heute weitgehend manuell durchgeführte Datenabgleich wird und soll zukünftig automatisiert erfolgen.
6. Die Datenmenge innerhalb des Unternehmens bleibt zukünftig konstant.
7. Die Befragung der Anwender belegt eine große Unzufriedenheit mit den bestehenden DV-Systemen im fertigungsnahen Umfelds, welche auf die dort zu findende unzureichende Integration und Vernetzung im EDV-Bereich zurückzuführen ist.
8. Die Wissenspropagierung stellt in heutigen Unternehmen einen Engpaß dar. Zukünftige Systeme sollen insbesondere das aktive Verbreiten des Wissens um Datenobjekte unterstützen.

Eine detailliertere Zusammenfassung und weitergehende Interpretation der Ergebnisse unter dem speziellen Blickwinkel der Anforderungen an integrierte Planungssysteme ist in [GKK95] zu finden.

3 Spiegelung der Anforderungen an den Arbeiten des Instituts

Die heute oft vorherrschende redundante Haltung der selben Daten an mehreren Stellen, zusammen mit dem allenfalls manuell angestoßenen Abgleich über dedizierte Filterprozeduren oder gemeinsame

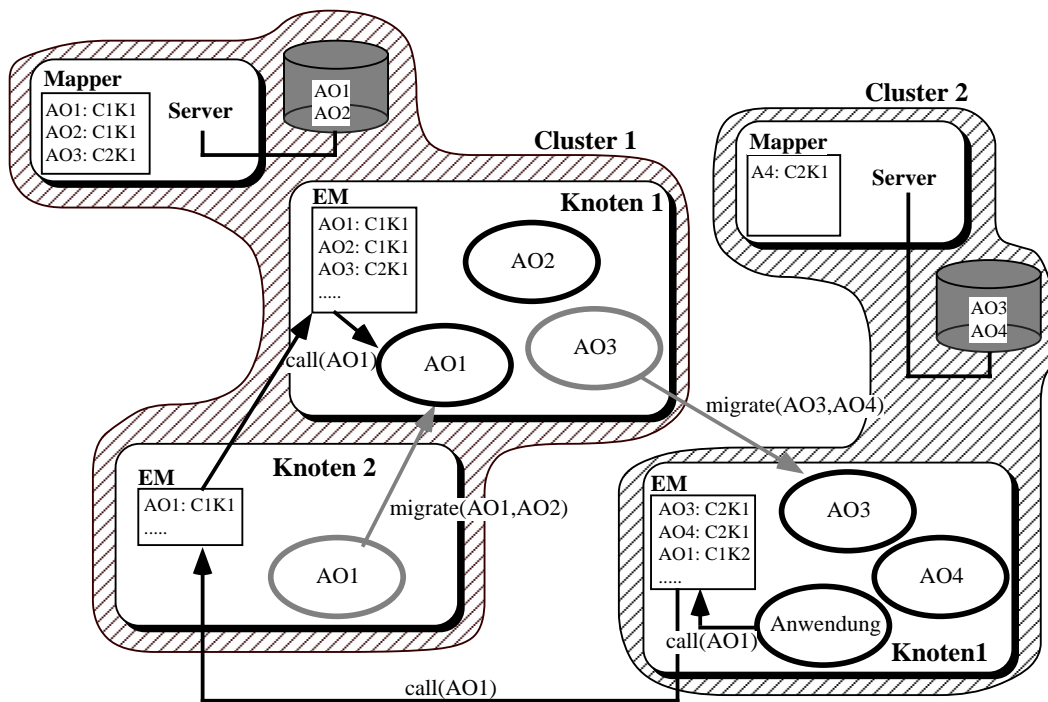


Abbildung 48. Cluster-Konzept

Datenmodelle, wie STEP, wurde als Status-Quo in der empirischen Studie immer wieder kritisiert. Verschärfend ergab sich, daß Toleranzen bzgl. der Ungenauigkeit oder der Veralterung der (unfreiwillig replizierten) Daten durch die immer stärker werdende Verzahnung von Abläufen zunehmend weniger tolerierbar sind.

Entsprechend dieser Erkenntnis gehen wir im folgenden von einer logisch zentralen Datenhaltung aus. Wesentlich ist hier die Erkenntnis, daß es sich um eine Zentralisierung auf logischer Ebene handelt. Eine direkte Abbildung auf eine physikalisch zentrale Datenhaltung in einem Rechnersystem — d.h. die geradlinige Umsetzung der Forderung nach einer logisch zentralen Datenhaltung — ist bei den vorzufindenden Rechnerinfrastrukturen verbunden mit den aktuellen Trends zum Rightsizing und Outsourcing kein gangbarer Weg. Der dabei auftretende Zwang, alle Datenzugriffe an eine zentrale Lokation zu leiten, ist allenfalls bei kleinen, auf einen Standort beschränkten Unternehmen gangbar. Eindeutige Argumente dagegen sind die Fehleranfälligkeit und die Kosten der zentralisierten Lösung. Folglich ist es unabdingbar, eine physikalisch verteilte Umgebung mit Arbeitsplatzrechnern, Workstations, mittlerer Datentechnik und Großrechnern als gegeben hinzunehmen und die logisch zentrale Lösung auf diese abzubilden. Wird bei dieser Abbildung trotz implizit vorhandener Datenreplikate für alle Benutzer der Eindruck erreicht, sie arbeiten mit nur einer Kopie der Daten, so spricht man von einer *One-Copy-View*.

Neben dem Wunsch nach einer logisch zentralen Datenhaltung existieren zwei weitere zentrale Rahmenbedingungen der integrierten Datenhaltung. Die erste ist, daß die Nutzungsstrukturen der Daten zwischen den einzelnen Nutzern oft komplex sind. So sind die Nutzerkreise der Daten immer eingeschränkt (z.B. wird der Geburtstag eines Kunden die Konstruktionsabteilung wenig interessieren) aber auch immer unregelmäßig überlappend (so interessieren den Leiter der Konstruktionsabteilung sowohl die Daten der Personalabteilung über seine Untergebenen, als auch die in seiner Abteilung erstellten Konstruktionen und die internen Auftragsbeziehungen seiner Abteilung). Die zweite Randbedingung ist die Änderung der Datennutzer und Nutzungsschwerpunkte zur Laufzeit. So verlagert sich der Nutzungsschwerpunkt von auftragsbegleitenden Daten mit dem Fortschritt der Auftragsbearbeitung im Unternehmen. Ein weiteres einfaches Beispiel ist die Verlagerung des Nutzungsschwerpunktes einer fertigen Konstruktion von den Workstations der Konstrukteure hin zur Arbeitsvorbereitung.

Eine Architektur, die die logisch zentrale Datenhaltung in einem verteilten System unterstützt, ist in dem in [KK94c] präsentierten und in Bild 48 skizzierten Cluster-Konzept zu finden. Dieses beruht auf der Gruppierung der gegebenen Datenverarbeitungs-Hardware nach administrativen Nutzungsgesichtspunkten unter Erhalt einer umfassenden Zugreifbarkeit auf jegliche gespeicherte Information innerhalb des Unternehmens. Das integrierte Konzept der Objektmigration wird der Forderung nach dynamischer Anpassbarkeit gerecht, dessen technische Einbettung in das Cluster-Konzept in [KK94b] vorgestellt wurde.

Diese zur Laufzeit flexible Systemarchitektur erlaubt es, die Modellierungsmächtigkeit des objektorientierten Datenmodells in eine verteilte Umgebung zu projizieren, um effizient eine logisch zentrale Sicht auf einen unternehmensweit verteilten Datenbestand zu erreichen [KKK⁺94].

Die Forderung nach einem unternehmensweiten Informationsverarbeitungsrückgrat ist mit der Bereitstellung einer integrierten objektorientierten Datenbank noch nicht vollständig Genüge getan. Der hierbei wesentlichste Punkt bei der Problemlösung ist die effiziente Auffindbarkeit von Informationen innerhalb der Datenbank. Alternativ spricht man auch vom Problem der Wissenspropagierung, wobei unter Wissen die Kenntnis über die in dem Datenverarbeitungssystem enthaltenen Daten zu verstehen ist. Klassischerweise sind hierfür in objektorientierten Datenbanken zwei Lösungskonzepte gebräuchlich [KK95]:

Die Anfrage an eine globale Typextension: Bei dieser Alternative wird die Suche nach Daten dadurch realisiert, daß die Anfrage an alle datenhaltenden Rechner weitergeleitet wird. Unmittelbar einsichtig ist, daß diese Strategie nur in kleinen Datenbeständen effizient umsetzbar ist, in großen, unternehmensweiten Beständen aber scheitern wird.

Die Verwendung persistenter Variablen: Bei dieser Alternative wird eine zum Zeitpunkt der Übersetzung der Anwendungsprogramme fixierte Abbildung eines anwendungsnahen Namens auf einen Datenbestand realisiert. Problematisch hierbei ist die Inflexibilität gegenüber dynamischen Änderungen der Erzeuger-/Konsumentenrelation auf den Daten.

Gelöst werden können diese Probleme durch die Trennung des Wissenspropagierungsmechanismus von den Datenhaltungskomponenten. Diese Idee wurde in [KK95] innerhalb des Kontextes des oben besprochenen unternehmensweiten Rückgrats nach [KK94c] [KK94b] präsentiert. Technisch beruht sie auf der Verwendung eines sogenannten Traders als Mechanismus zur dynamischen Abbildung von Benutzerspezifikationen auf einen Datenbestand. Hierdurch ist es möglich, die Wissenspropagierung an aktuelle Änderungen sehr flexibel anzupassen, ohne die betroffenen Programme verändern bzw. neu übersetzen zu müssen. Ferner ist eine derartige Anpassung nicht nur an den Stellen möglich, an denen eine derartige Änderung zum Zeitpunkt des Entwurfes des Softwaresystems bereits vorhergesehen wurde, sondern an beliebigen Stellen, die den Wissenspropagierungsmechanismus verwenden.

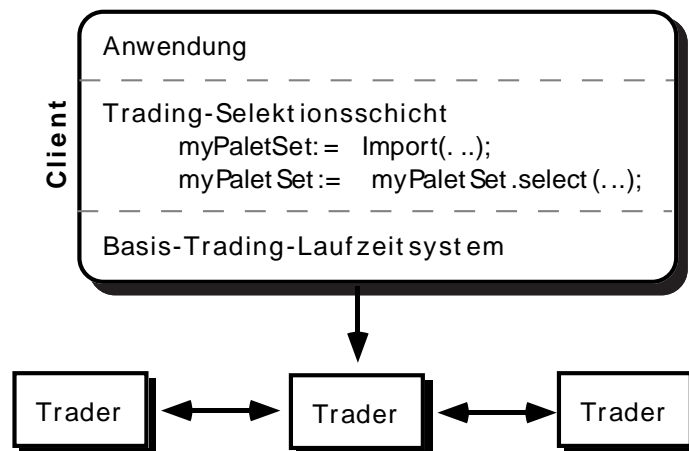


Abbildung 49. Schichtung Endsystem

Im Gegensatz zu den oben zu findenden Lösungen weist dieser Ansatz in seiner Urform den Nachteil nicht-atomarer Operationen auf, so daß sich aus Benutzersicht schwer durchschaubare Anomalien ergeben können. Beispielsweise ist eine Löschoperation nicht unteilbar gestaltet, so daß es bei der Wissenspropagierung zu unerwünschten Race-Conditions kommen kann. Beseitigt werden kann dieser Nachteil durch eine zweite, in den Anfragemechanismus der Anwenderseite der Datenbankschnittstelle integrierte Selektionsschicht mit Datenbankeinbindung, die in [KK94a] vorgestellt wurde. Insgesamt ergibt sich eine Schichtung des Laufzeitsystems eines Datenbank-Clients, wie in Abb. 49 dargestellt. Einen Überblick über die Eigenschaften der Alternativen gibt Tabelle 8.

Zusammenfassend ergibt sich durch das Zusammenspiel der zur Laufzeit flexiblen Systemarchitektur mit den ebenfalls dynamischen Wissenspropagierungsmechanismen ein System, das den modernen, stark verzahnten Abläufen in verteilten aber dennoch integrierten Systemen gerecht wird.

	Typextension	Peristente Variable	Trader	Trader mit Selektionsschicht
Effizienz	—	•	•	•
Flexibilität	•	—	•	•
Homogene Sprachintegration	0 ^a	•	—	0 ^b
Aktualität	•	•	—	•
Konsistent	•	•	—	•

Tabelle 8. Übersicht über mögliche Wissenspropagierungsmechanismen

^a Durch die Notwendigkeit einer expliziten objekt-orientierten Query-Sprache

^b Durch die Notwendigkeit, die Selektionsschicht als Indirektionsstufe zu benutzen.

4 Zusammenfassung

In diesem Beitrag wurden die Anforderungen moderner ingenieurwissenschaftlicher Anwendungen an integrierte Informationssysteme diskutiert. Neben den möglichen Einbindungen von Datenbanken wurde ein System lose-gekoppelter Cluster vorgestellt, welches der Anforderung nach einer logischen Zentralisierung des Datenbestandes unter der aus Effizienz- und Kostengründen unabdingbaren physikalischen Verteilung gerecht wird. Dieses System zeichnet sich in erster Linie durch seine Fähigkeit zur dynamischen Adaption an geänderte Anforderungsprofile aus.

In weiteren Arbeiten soll die softwaretechnische Einbindung des Trading-Systems in erweiterten Anwendungen untersucht und ein integrierter Entwicklungsprozeß, der Trading- und Datenbankschema zusammen umfaßt, untersucht werden. Zudem wird aktuell untersucht, wie die Kooperation mehrerer Instanzen über die Datenbank durch ein höheres Kommunikationskonstrukt — die Allianzen — unterstützt werden kann, und wie diese zur Realisierung benutzernaher Migrationspolitiken [KLW95] eingesetzt werden können. Schließlich werden Verteilungserweiterungen basierend auf modernen, kommerziellen Laufzeitsystemen wie Objectstore implementiert. Insgesamt stehen die Arbeiten unter einer engen Kooperation mit der Gruppe von Herrn Lockemann am IPD und einer Gruppe am wbk, wobei speziell die Nutzung der Infrastruktur am PTL (Produktionstechnischen Labor) im Vordergrund steht.

CogPiT — Configuration of Protocols in TIP

Burkhard Stiller¹⁴

1 Project Partners

This report presents briefly the work done within the SIEMENS-project called **CogPiT** (Configuration of **P**rotocols in **T**IP). The Institut für Telematik, Universität Karlsruhe agreed upon a cooperation contract with SIEMENS, Zentrale für Forschung und Entwicklung (ZFE) in Neu-Perlach/München to develop a configuration approach for communication protocols within the SIEMENS-specific run-time and protocol development environment TIP (Transport and Internetworking Package).

2 Project Targets

The specific target of the project **CogPiT** is the definition and implementation of a configuration task for protocols, especially for the TEMPO protocol as one example. Prerequisites for this task are requested QoS requirements of an application and the prevailing status of the underlying network and the run-time system. The result of this configuration task is an application-tailored communication protocol, which is represented as a module configuration, including configured protocol mechanisms and their interconnections. The second task of downloading these module configurations into the TIP environment (Channels) and running the executable protocol is already part of TIP and has been described elsewhere. Therefore, this overview will briefly outline components of the required environment, only.

3 Introduction

Existing communication protocols and processing environments are not well suited for a sufficient support of innovative applications, their emerging Quality-of-Service (QoS) requirements, and future networks, since the protocol functionality of considered protocols is limited and pre-defined and the run-time systems are not efficient. Nonetheless, QoS has to be provided by the underlying communication subsystem and the network to allow for the sufficient processing of application-supporting protocols. Especially, standard protocols offer a rich but fixed set of protocol functionality that is not always acceptable for certain application requirements. *E.g.*, the standard ISO/OSI Transport Protocols, such as TP4 and its service specification [ISO86], [ISO88], neither provide a jitter control functionality nor a jitter QoS parameter to support isochronous traffic adequately. Additionally, since traditional models of protocol processing do not show a well suited behavior [Svo89], [HP91], solutions for achieving a better behavior have to be developed.

4 TIP — The Transport and Internetworking Environment

As the requested service requirements increase, a new transport protocol and architecture is needed, to satisfy all new services. Therefore, the protocol and architecture called *Transport and Internetworking Package (TIP)* has been developed [BV93]. TIP consists out of the run-time system Channels, the transport protocol TEMPO, and Netglue, a subnet convergence sublayer, besides the developed *Configuration Algorithm COFAL*.

Therefore, TIP can be configured using the configuration task developed for this project to provide a specifically requested service according to application demands (such as from video conferencing, traditional FTP, or image browsing) and network prerequisites given by, *e.g.*, ATM [R. 94], FDDI [Ros86], or Ethernet. The scalable functionality of TIP is its main advantage. Newly defined building blocks may be added to the system in the future to support modern application characteristics. Besides TEMPO, well-known protocols, such as TCP (Transmission Control Protocol) [DAR81b] or IP (Internet Protocol) [DAR81a], may be used on top of any of the mentioned networks without any changes to the TIP environment.

¹⁴ The author is currently on leave from Universität Karlsruhe, Institut für Telematik, D-76128 Karlsruhe, Germany, and is sponsored by the Commission of the European Communities as a Research Fellow under the Human Capital and Mobility Scheme (RG 19372) at the University of Cambridge, Computer Laboratory, Systems Research Group, New Museums Site, Pembroke Street, Cambridge CB2 3QR, England, U.K. This present report is an excerpt of the final project report to be delivered.

A closer look onto TIP unveils a control and user plane, which are responsible for the set-up and release of TIP associations and the user data transfer, respectively. Especially for user data transfer the requested communication protocol is configured, but associations with different degrees of security may be configured as well. Furthermore, associated set-ups and releases may be supported by certain security functionality that can be configured as well.

4.1 The Run-Time System Channels

Channels is the run-time system for protocol processing within TIP [BS94a]. It is supposed to be a part of TIP, providing an efficient support of multi-media applications. In order to design an efficient run-time system for protocol processing the following *objectives* have been taken into consideration:

- Providing a basis for the portability of communication protocols.
- Supporting extensions of existing communication protocols within the same environment.
- Definition of a configurable architecture allowing for the execution of protocol building blocks of different levels of granularity.
- Parallel processing of independent protocol building blocks is required.
- Supporting the real-time capability of modern applications.
- Migration of protocol building blocks into hardware components shall be possible.
- Providing a simple method for developing communication protocols based on libraries and basic protocol operations.
- Allowing for a performance-oriented design applying threads and light weight context switches.

The Architecture of Channels encompasses light weight threads and libraries for communication purposes, such as for timer or messages. The service of Channels is used by clients, such as the user or the network driver. Multiple modules — defining the protocol processing functionality — may be interconnected to form the requested protocol. Standardized messages are exchanged between these modules to communicate between them. Upcalls and downcalls are processed between a module and its successor and predecessor respectively. Events and control messages are exchanged between child and parent modules, which is used for creating and deleting modules in the run-time system. Finally, an entire collection of specific modules that have been configured according to application needs, is described by three different *channel templates*: the “channel configuration”, the “module configuration”, and the “run-time configuration”.

4.2 TEMPO — A Transport Protocol

The transport and internetworking protocol TEMPO includes the specification of some protocol mechanisms that are useful to support various application requirements [Bø93], [BV93]. In summary, these selected mechanisms form a lightweight protocol. Therefore, TEMPO offers a transport service to applications that is configurable as well.

QoS Structure in TEMPO — Quality-of-Service (QoS) is used to specify application requirements that are supported by transport associations within TIP. These transport associations model the logical binding between users. The manner of communication within these types of associations is defined by transport streams that allow for the unidirectional flow of transport service data units (TSDU) from the source (sender) to one or multiple sinks (receiver).

The quality of a single stream may be specified by valid QoS parameters that are defined in the *QoS structure* of TEMPO. Therefore, multiple streams being part of one association may have different QoS values assigned to. As depicted in some examples in Table 9, 43 different QoS parameters are directly supported by TEMPO, 8 others have been added for the CogPiT project according to a wider range of application specification possibilities. *E.g.*, the *workahead* parameter has different meanings for different types of streams. In a continuous stream *workahead* of the requesting side equals the burstiness, which is the maximum number of TSDUs passed to the provider before their actual time slot. For the indicating side it defines the maximum number of TSDUs accepted from the service provider before their actual indication time. An asynchronous stream regards the *workahead* parameter as the definition of the window size for the flow control function. Especially, if the formal parameter, called symbolic variable, *DSQ_Workahead* equals zero, than an isochronous type of traffic has been specified, which means that no jitter is allowed — or at least a certain maximum may not be exceeded. The maximum throughput of an application may be calculated within the TEMPO QoS structure, using the interval parameter and the specification of the maximum TSDU length.

QoS Parameter Name	Description	Unit
	<i>Guarantee</i>	
DSB_Best_Effort	QoS values provided best as possible	bool
DSB_Soft_Guarantees	QoS values provided under soft conditions	bool
DSB_Hard_Guarantees	QoS values provided under worst conditions	bool
	<i>Delay</i>	
DSQ_Delay_Max	Max. Delay between T-Data.req and T-Data.ind	ms
DSQ_Delay_Min	Min. Delay between T-Data.req and T-Data.ind	ms
	<i>Interval</i>	
DSQ_Interval	Time between two successive T-Data.req or T-Data.ind, which are equal.	ms
	<i>Workahead</i>	
DSQ_Workahead	(cf. text)	byte
	<i>Size</i>	
DSQ_Max_TSDU_Length	Maximum number of octetts of a TSDU	byte
	<i>Examples for Damage</i>	
DSQ_Bit_Error_Rate	Residual Bit Error Rate	neg. dec. log.
DSB_Damage_Ignore	No detection and correction of damage	bool
DSA_Checksum_CCITT	Use CCITT CRC	bool
DSA_FEC	Use Forward Error Correction	bool
	<i>Examples of Loss</i>	
DSB_Loss_Ignore	No loss detection and correction	bool
DSB_Loss_Notify	Loss detection and notification for detected losses	bool
	<i>Examples for Confidentiality</i>	
DSA_DES_Con	Use Data Encryption Standard (DES)	bool
DSA_Triple_DES_Con	Use Triple-DES-algorithm	bool

Tabelle 9. Examples for QoS Parameters of TEMPO, their Symbolic Variables, and Units

Functions and Mechanisms in TEMPO — Within the TEMPO protocol certain functionality is defined, which encompasses, *e.g.*, segmentation, error control, checksumming, acknowledgements, and flow control. Table 10 includes in an overview five different *uses* of functions and mechanisms in CogPiT, while the different uses depict the specification’s origin of the mechanisms, such as the definition within TEMPO (*T*), the functional composition of [M. 94] (*C*), current used extensions for configuration purposes (*X*) and future use (*F*) as well.

TEMPO is used for configuration purposes within TIP. Therefore, the mentioned decomposition of the protocol into separate building-blocks is necessary. Relying on these fundamentals [M. 94], the following decomposition of TEMPO, which is useful for configuring different protocols, while distinguishing certain mechanisms (such as various checksumming alternatives), has been developed and specified. *E.g.*, Figure 50 depicts a mechanism/module-grained view in a graphical manner, while the management (MGT), service access point (SAP), and association end point (AEP) functionality of TIP are introduced already. The semantics of “mechanisms” as such (nodes) and “interconnections” (arrows), defining the successor/predecessor relation of mechanisms, are used for configuration purposes.

5 The Task of Configuration

Each configurable basic building block — called module, independent of its granularity — has to be specified in terms of relevant information that are important for (1) the configuration task and (2) the instantiation task. The former task of configuration includes the selection of necessary and required modules out of a pool of modules. The second task of instantiation maps the executable modules onto the underlying hardware platform. CogPiT adapts the approach of the *Protocol Configuration Manager (PROCOM)* [Sti94b] approach of the Function-based Communication Subsystem F-CSS [ZST93] to the special TIP environment [BS94a].

In the TIP environment the configuration task remains principally the same as within PROCOM except for the following enhancements due to TIP. The introduction of additional TIP-specific configuration information and the separation of successor/predecessor information for mechanisms is required. The dedicated TIP run-time system Channels has to be taken care of with certain network and system

Use	Protocol Function	Protocol Mechanism
C T	Segmentation	PDU_Segmentation
C T	Reassembly	PDU_Reassembly
C T	PDU_Composition	PDU_Composition_TEMPO
C T		Reply_Composition_TEMPO
C T	Checksum_Calculate	Checksum_Calculate_Data_OSI
C T		Checksum_Calculate_Data_CCITT
C T		Checksum_Calculate_Data_ANSI
C T		Checksum_Calculate_Data_32
F		Checksum_Calculate_Header
C T	Checksum_Validate	Checksum_Validate_Data_OSI
C T		Checksum_Validate_Data_CCITT
C T		Checksum_Validate_Data_ANSI
C T		Checksum_Validate_Data_32
F		Checksum_Validate_Header
C T	Flow_Control—Sender	Flow_Control_rate_based_Sender
X		Flow_Control_window_based_Sender
C T	Flow_Control_Receiver	Flow_Control_rate_based_Receiver
X		Flow_Control_window_based_Receiver
F		Stop_and_Go
C T	Error_Control	Error_Control_TEMPO
F		Discard_PDU
C T	Check_PDU	Check_PDU_TEMPO
C T	Sink_Status	Sink_Status_TEMPO
C T	Acknowledgement	Acknowledgement_selective
X		Acknowledgement_cumulative
F	Retransmission	Go_Back_N
F		Automatic_Repeat_Request
T		Timer_Driven_Retransmission
X		Retransmission_selective
C T		Retransmission_cumulative
T		Forward_Error_Correction
F	Synchronization	Inter_Stream_Synchronization
F		Intra_Stream_Synchronization

Tabelle 10. Considered Protocol Functionality

resource parameters that are specifiable in a formal language for each module. This is necessary for providing information for the second step of the protocol configuration instantiation, only. Therefore, parameters such as the type of a module, parent and childs of modules have been introduced. Finally, the resulting output corresponds to Channels requirements.

Quality-of-Service as well as Network and System Resource Parameters — Application requirements have to be expressed within particular parameters. These QoS parameters (cf. Table 9) are used to stimulate the configuration task, while according to particular requirements certain protocol mechanisms may be selected or not. Therefore, QoS parameters in terms of their symbolic variables guide the configuration task. TIP environmental issues are described using Network and System Resource (NSR) parameters. These parameters include information on the current state of the network and the end-system. In the CogPiT case they are maintained by the run-time system Channels, since it stores all relevant information, gaining access in a formalized manner for the configuration task.

Module Description Language — For configuration purposes protocol functions and mechanisms have to be described as separate modules in a formal language. The formal language used in this case is called *Module Description Language (MDL)*. This description involves conditions that define exactly, under which circumstances a module can be configured as part of a communication protocol. Furthermore, information containing interdependencies between modules and environment-specific data as well have been added. These include, *e.g.*, input and output parameters, additional information on user, child, and provider modules, and the type of modules.

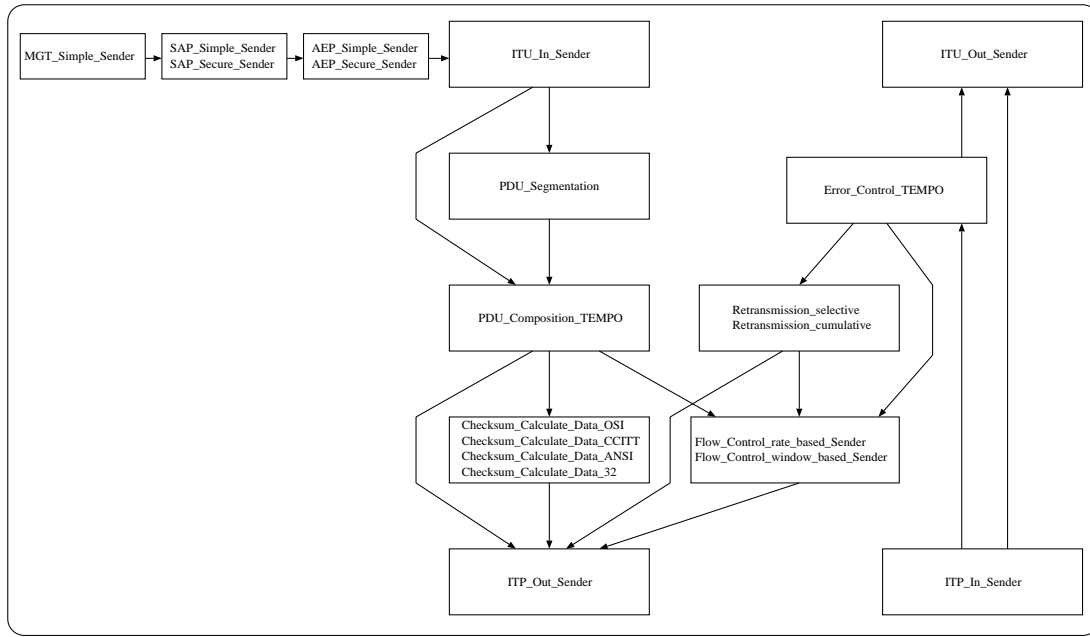


Abbildung 50. Sender Mechanisms of TEMPO (Source)

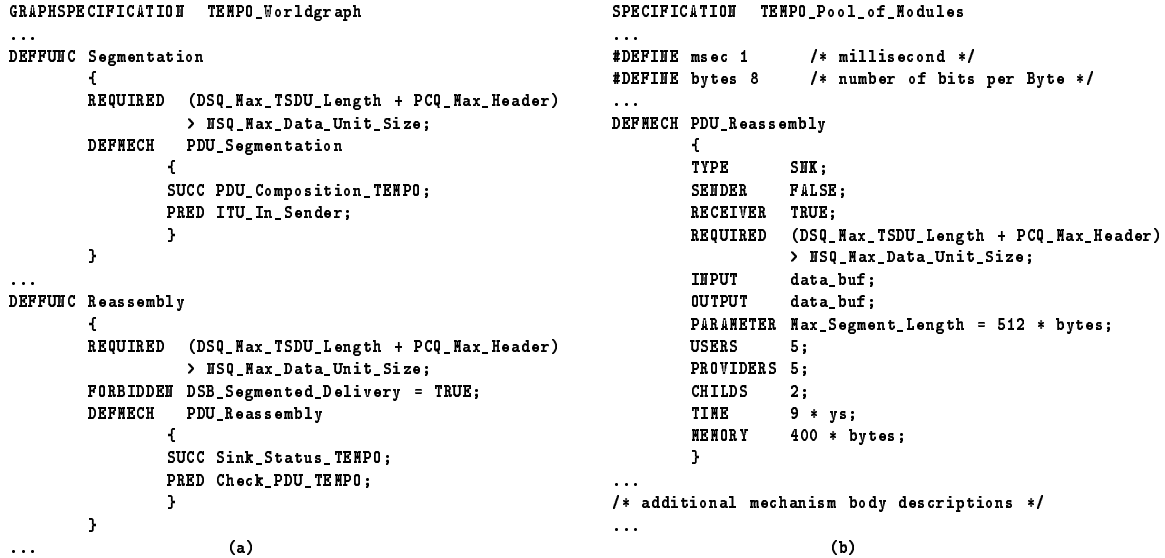


Abbildung 51. An Extract of a GRAPHSPECIFICATION and a SPECIFICATION

Firstly, the GRAPHSPECIFICATION includes the specification of the *worldgraph*, which contains besides others the indispensable successor/predecessor information for modules. This relation allows for the correct specification of processing the configured modules in the right order. An example can be seen in Figure 51 (a).

Secondly, the SPECIFICATION includes details on all modules — the protocol mechanisms — in the *pool of modules*. This part of the MDL specification can be maintained naturally and derived automatically, since only module internal details are included and no module interconnections are considered. A short example is included in Figure 51 (b).

5.1 The Configuration Task

As depicted in Figure 52, the overall view of the configuration task has been summarized in a graphically manner. The inputs to the *Configuration Algorithm COFAL* are drawn on the left side, such as the QoS parameter, the NSR parameter, the pool of modules, and the worldgraph. Depending on these

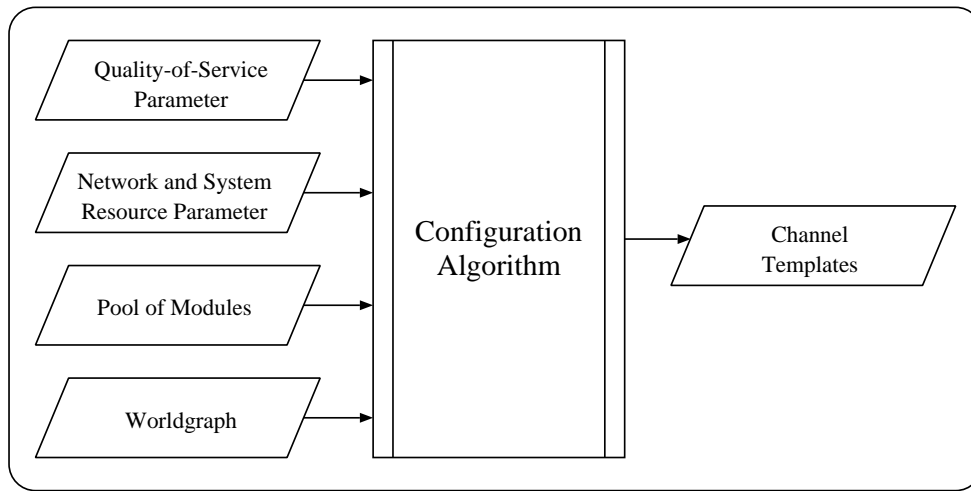


Abbildung 52. The Configuration Task in an Overview

information, channel templates, drawn on the right side, are generated, representing the configured communication protocol.

COFAL needs information on application requirements and on resources, which are handed over to COFAL by the application and the local run-time system. Furthermore, COFAL is dependent on the formal specified pool of modules and its included protocol function and mechanism specifications. The task of configuration itself is independent of any underlying hardware architecture used to execute the communication protocol, since the result of a COFAL configuration is a *channel template* — a formal representation of a configured protocol, *e.g.*, consisting out of multiple protocol mechanisms.

According to various separate features to be checked, COFAL is designed by eight separate steps. These steps are called as follows:

- Step 1 — Configuration Preparation
- Step 2 — Determination of Available Resources
- Step 3 — Feasibility Check
- Step 4 — Selection of Protocol Functions and Mechanisms
- Step 5 — Protocol Configuration Construction
- Step 6 — Protocol Configuration Consistency Check
- Step 7 — Parameter Evaluation
- Step 8 — Generation and Consistency Check of Channel Templates

Especially step 1 to 4 are applied to the MDL specification resulting in two intermediate sets of selected mechanisms, one for the sender's side the other one for the receiver's side of the configured communication protocol. Steps 5 to 8 are applied afterwards to each of these two sets separately to allow for the correct calculation for the final channel templates.

6 Summary and Conclusions

The configuration algorithm has been developed according to TIP-specific requirements. Relevant prerequisites of the TIP environment have been taken into account and a prototype has been implemented for the SUN platform.

Concluding, the project targets have been achieved in terms of a complete definition of protocol functionality, an extensive classification of protocol functions, an adequate specification of environmental information (QoS and NSR), a sufficient specification of configuration conditions, and an independence of the configuration algorithm from the specified functionality. The demonstration of the achieved results will take place this year at the end of March in München.

Acknowledgements: Many thanks are addressed to Joachim Elstner and Kai Schubert, who provided lots of their technical skills for porting, adapting, and testing the configuration algorithm at the SUN platform. Additionally, Martina Zitterbart managed the local affairs of the project in Karlsruhe, while the author is working on a PostDoc position at the University of Cambridge, England.

Anhang

Literatur

- [ADH⁺92] M. Altenhofen, J. Dittrich, R. Hammerschmidt, R. G. Herrtwich, T. Käppner, C. Kruschel, A. Kückes und T. Steinig. The BERKOM Multimedia Collaboration Service. DeTeBerkom GmbH, Oktober 1992.
- [AKTZ94] J. M. Abler, O. G. Koufopavlou, A. N. Tantawy und M. Zitterbart. On the Design of a Multigigabit Router. *Journal on High Speed networks* 3(3), 1994, Seite 209 – 232.
- [AL93] M. B. Abbott und Peterson L. L. Increasing Network Throughput by Integrated layer Processing. In *IEEE/ACM Transactions on Networking*, Vol. 1, No. 5, Oktober 1993, Seite 600–610.
- [Arm94] G.J. Armitage. gNET: An ATM LAN Signalling Protocol. Technischer Bericht ISBN 85867 077-1, University of Melbourne, Department of Electrical Engineering, Grattan Street, Parkville 3052, Melbourne, Australia, February 1994.
- [Ash93] C. Ashford. Comparison of the OMG and ISO/CCITT Object Models. Technischer Bericht, Telecom Canada, Dept. 6L53, P.O. BOX 3511, Station C, Ottawa Ontario K1Y 4H7 Canada, April 1993.
- [ATM93] ATM-Forum. *ATM User Network Interface Specification, Version 3.0*. Prentice Hall, Englewood Cliffs, New Jersey, U.S.A. 1993.
- [ATM94a] ATM-Forum. *ATM User Network Interface Specification, Version 3.1*, 21. July 1994.
- [ATM94b] ATM-Forum 94-0998. *Signalling Subgroup Meeting Notes, UNI 4.0 Features*, 26. – 29. September 1994.
- [Bal86] M.O. Ball. Computational Complexity of Network Reliability Analysis: An Overview. *IEEE Transactions on Reliability* R-35(3), August 1986, Seite 230–239.
- [BB93] Stewart F. Bryant und David L.A. Brash. The DECNIS 500/600 Multiprotocol Bridge/Router and Gateway. Digital Technical Journal 5, Digital Equipment Corporation, 1993.
- [BD95] Torsten Braun und Christophe Diot. Protocol Implementation Using Integrated Layer Processing. ACM Sigcomm '95, Cambridge, Massachusetts (USA), 30. August - 1. September, 1995.
- [BDG91] R. Bubenick, J DeHart und M. Gaddis. Multipoint Connection Management in High Speed Networks. In *IEEE INFOCOM 1991*, Bal Harbor, Florida, U.S.A., 9.-11. April 1991. Seite 59–68.
- [BDH95] Torsten Braun, Christophe Diot und Anna Hoglander. An Experimental User-Level Implementation of TCP. *interner Bericht*, 1995.
- [Bec93] J. Becher. *Konfigurierung verteilter, heterogener Informationsverarbeitungssysteme*. Verlag Shaker. 1993.
- [Ber93] Niels Ole Bernsen. Specificity and Focus: Two Complementary Aspects of Analogue Graphics and Natural Language. Technischer Bericht, Roskilde University, Denmark, 1993. Esprit Basic Research Project GRACE Deliverable 2.1.3.
- [Bø93] S. Böcking. *TEMPO – Transport Protocol Specification*. München, Germany, 1. October 1993.
- [Bou91] T. Bouron. What Architecture for Communications Among Computational Agents? In *AAAI'91 Workshop on Cooperation Among Heterogeneous Intelligent Systems*, Anaheim, CA, 1991.
- [Bra93] Braun. *Ein paralleles Transportsubsystem für zellenbasierte Hochgeschwindigkeitsnetze*. Dissertation, Universität Karlsruhe, 1993.
- [Bra94a] Torsten Braun. Architektur, Implementierung und Leistungsbewertung eines parallelen Transportsubsystems für zellenbasierte Hochgeschwindigkeitsnetze. In *GI/ITG-Fachgruppe 'Kommunikation und Verteilte Systeme': 1. Arbeitstreffen 'Architektur und Implementierung von Hochleistungs-Kommunikationssystemen'*, Karlsruhe, 17./18. Januar 1994.
- [Bra94b] Torsten Braun. PATROCLOS: A Flexible and High-Performance Transport Subsystem. In *4th IFIP Workshop on Protocols for High-Speed Networks*, Vancouver, 10.-12. August 1994. Seite 183–201.
- [Bra95a] S. Bradner. RFC 1752: The Recommendation for the IP Next Generation Protocol, Januar 1995.
- [Bra95b] Torsten Braun. Ein paralleles Transportsubsystem für zellenbasierte Hochgeschwindigkeitsnetze. In *Kommunikation in Verteilten Systemen*, Informatik aktuell, Chemnitz, 22.24. Februar 1995. Springer-Verlag, Seite 562–566.
- [Bra95c] Torsten Braun. Limitations and Experiences of Integrated Layer Processing. In *1. Gemeinsame Jahrestagung der Gesellschaft für Informatik (GI) und der Schweizer Gesellschaft für Informatik (SI)*, Zürich, 18.-20. September 1995.
- [BS94a] S. Böcking und V. Seidel. TIP's Protocol Run-Time System. In *12th Annual Conference on European Fiber Optic Communications and Networks, EFOC&N 94*, Heidelberg, Germany, 21. – 23. June 1994. Seite 145–149.
- [BS94b] Torsten Braun und Claudia Schmidt. A Parallel Transport Subsystem Implementation for High Performance Communication. *Concurrency: Practice and Experience, Special Issue on High Performance Distributed Computing*, Juni 1994, Seite 375–391.
- [BSSZ94] Torsten Braun, Claudia Schmidt, Burkhard Stiller und Martina Zitterbart. A Protocol Architecture for a Flexible High Performance Communication Subsystem. In *CaberNet Workshop*, Dublin, 23.-25. März 1994.
- [BSZ94a] Torsten Braun, Jochen Schiller und Martina Zitterbart. Implementation of Transport Protocols Using Parallelism and VLSI Components. In *International Telecommunication Symposium 1994 (ITS'94)*, Rio de Janeiro, 22.-25. August 1994. Seite 285–290.

- [BSZ94b] Torsten Braun, Jochen Schiller und Martina Zitterbart. A Modular VLSI Implementation Architecture for Communication Subsystems. In *4th IFIP Workshop on Protocols for High-Speed Networks*, Vancouver, 10.-12. August 1994. Seite 119-134.
- [BV93] S. Böcking und P. Vindeby. TIP: A Transport and Internetworking Package for ATM. 1993.
- [BY92] Balraj und Yemini. Putting the Transport Layer on VLSI - the PROMPT Protocol Chip. IFIP, North-Holland, 1992, Seite 19-34. *Protocols for High-Speed Networks III*.
- [BZ93] Braun und Zitterbart. Parallel Transport System Design. 1993, Seite 397-412.
- [BZ94] Torsten Braun und Martina Zitterbart. Videokonferenzen im Internet. In *Praxis der Informationsverarbeitung und Kommunikation (PIK)* 17, 4/1994, 1994, Seite 253-254.
- [Cal92] R. Callon. RFC 1347: TCP and UDP with Bigger Addresses (TUBA), A Simple Proposal for Internet Addressing and Routing, Juni 1992.
- [Car92] Niels V. Carlsen. Towards a Common Context for User Interface Management System Design. In J. Larson und C. Unger (Hrsg.), *Engineering for Human-Computer Interaction*, IFIP Transactions, Ellivuori, Finland, August 10-14 1992. IFIP, North-Holland Amsterdam, Seite 13-32. Proceedings of the IFIP TC 2/WG 2.7 Working Conference.
- [Car94a] Georg Carle. Adaptation Layer and Group Communication Server for Reliable Multipoint Services in ATM Networks. In R. Steinmetz (Ed.): *Advanced Teleservices and High Speed Communication Architectures; Lecture Notes in Computer Science 882*, Springer-Verlag, S. 124-138, Second International Workshop, IWACA '94, Heidelberg, September 1994.
- [Car94b] Georg Carle. Einsatz von Übertragungswiederholung und Vorwärtsfehlerkorrektur für zuverlässige Multicastkommunikation in ATM-Netzen. In *ITG Fachtagung Codierung für Quelle, Kanal und Übertragung*, München, Oktober 1994.
- [Car94c] Georg Carle. Error Control for Reliable Multipoint Communication in ATM Networks. In *Third International Conference on Computer Communications and Networks ICCCN'94*, S. 33-37, San Francisco, CA, U.S.A., September 1994.
- [Car94d] Georg Carle. Reliable Group Communication in ATM Networks. In *Twelfth Annual European Fibre Optics Communications and Networking Conference EFOCN'94*, S. 30-34, Heidelberg, Juni 1994.
- [Car95] Georg Carle. Provision of Reliable Multipoint Services in ATM Networks. In *International Conference on ATM Developments '95*, Rennes, Frankreich, März 1995.
- [CCG⁺93] A. Campbell, G. Coulson, F. Garcia, D. Hutchinson und H. Leopold. Integrated Quality of Service for Multimedia Communications. In *IEEE INFOCOM'93*, San Francisco, April 1993. Seite 732 - 739.
- [CCI89] CCITT. *Recommendation Z.100*, 1989.
- [CD94] K. Cox und J. DeHart. Connection Management Access Protocol (CMAP) Specification, Version 3.0. Technischer Bericht WUCS-94-21, Department of Computer Science, Washington University, St. Louis, Missouri, U.S.A., 10. November 1994.
- [CFSD90] J.D. Case, M. Fedor, M. Schoffstall und J.R. Davin. A Simple Network Management Protocol (SNMP). Network Working Group, Request for Comments 1157, Mai 1990.
- [CH94] Jon Crowcroft und Christian Huitema. *1st International Workshop on High Performance Protocol Architectures (HIPPARCH'94)*. Sophia-Antipolis. 15.-16. Dezember 1994.
- [CL92] Norman Carver und Victor Lesser. The Evolution of Blackboard Control Architectures. Technischer Bericht 92-71, CMPSCI, 1992.
- [Coc90] Gilbert Cockton. The Architectural Bases of Design Re-use. In D.A. Duce, M.R. Gomes, F.R.A. Hopgood und J.R. Lee (Hrsg.), *User Interface Management and Design*, Eurographic Seminars, Lisbon, Portugal, Juni 4-6 1990. Springer-Verlag, Seite 16-34. Proceedings of the Workshop on User Interface Management Systems and Environments.
- [Cou87] Joelle Coutaz. PAC, an Object Oriented Model for Dialog Design. In H.-J. Bullinger und B. Shackel (Hrsg.), *Human-Computer Interaction — INTERACT'87*, Stuttgart, Germany, September 1-4 1987. IFIP, Elsevier Science Publishers, Seite 431-436.
- [Cro93] S. Crosby. MSNL Connection Management. Technischer Bericht ATM Document Collection 3 (Blue Book), University of Cambridge, Computer Laboratory, Cambridge, England, U.K., 16. February 1993.
- [Cro94] S. Crosby. MSNL UNI Signalling Protocol, DRAFT Specification. Technischer Bericht Revision 1.5, University of Cambridge, Computer Laboratory, Cambridge, England, U.K., 1. December 1994.
- [CS93] David R. Cheriton und Dale Skeen. Understanding the Limitations of Causally and Totally Ordered Communication. In *Proc. 13th ACM Symposium on Operating Systems Principles*. ACM, December 1993, Seite 44-57.
- [CS95] Georg Carle und Jochen Schiller. Modular VLSI Implementation Architecture for the Provision of High-Performance Multipoint Multimedia Services. In *Silicon Valley Networking Conference SV-NC95*, San Jose, CA, April 1995.
- [CSS94] Georg Carle, Jochen Schiller und Claudia Schmidt. Support for High-Performance Multipoint Multimedia Services. In D. Hutchison, A. Danthine, H. Leopold, G. Colson (Eds.): *Multimedia Transport and Teleservices; Lecture Notes in Computer Science 868*, Springer-Verlag, S. 219 - 240, International COST 237 Workshop, Wien, Österreich, November 1994.

- [CT90a] Clark und Tennenhouse. Architectural Considerations for a New Generation of Protocols. ACM, September 1990, Seite 200–208. SIGCOMM '90.
- [CT90b] David Clark und Dave Tennenhouse. Architectural considerations for a new generation of protocols. In *ACM SIGCOMM*, September 1990, Seite 200–208.
- [CZ95] Georg Carle und Martina Zitterbart. ATM Adaptation Layer and Group Communication Servers for High-Performance Multipoint Services. In *7th IEEE Workshop on Local and Metropolitan Area Networks*, Marathon, Florida, U.S.A., März 1995.
- [DAR81a] DARPA. *Internet Protocol — DARPA Internet Protocol Specification, RFC 793*, September 1981.
- [DAR81b] DARPA. *Transmission Control Protocol — DARPA Internet Protocol Specification, RFC 791*, September 1981.
- [DGB92] J. DeHart, M. Gaddis und R. Bubenick. Connection Management Access Protocol (CMAP) Specification, Version 2.1.1. Technischer Bericht WUCS-92-01, Department of Computer Science, Washington University, St. Louis, Missouri, U.S.A., 7. May 1992.
- [DPD94] Druschel, Peterson und Davie. Experiences with a High-Speed Network Adaptor: A Software Perspective. ACM, August 1994. SIGCOMM '94, London.
- [Dre94] Stefan Dresler. Komponenten zur Unterstützung zuverlässiger Gruppenkommunikation in ATM-Netzen. In *Diplomarbeit, Institut für Telematik, Universität Karlsruhe*, Oktober 1994.
- [DSW93] G. Dück, T. Scheuer und H.M. Wallmeier. Toleranzschwelle und Sintflut: neue Ideen zur Optimierung. *Spektrum der Wissenschaft*, März 1993, Seite 42–51.
- [DV92] J. M. Duran und J. Visser. International Standards for Intelligent Networks. *IEEE Communications Magazine*, February 1992, Seite 34 ff.
- [ETS92a] ETSI Standard, DE/SPS 5024. *Signalling: Basic Call*, 1992.
- [ETS92b] ETSI Standard, DE/SPS 5034. *Signalling: Supervisory Call*, 1992.
- [Fel94] Feldmeier. *High Performance Communication*, Kapitel An Overview of the TP++ Transport Protocol. Kluwer Academic Publishers. 1994.
- [FGM94a] Oliver Frick, Hans-Werner Gellersen und Max Mühlhäuser. Developing Cooperative Media-Integrated Software. In Bernd Wolfinger (Hrsg.), *Innovationen bei Rechen- und Kommunikationssystemen*, Hamburg, August 1994. 24. GI-Jahrestagung, Springer.
- [FGM94b] Oliver Frick, Hans-Werner Gellersen und Max Mühlhäuser. Multiuser and Multimodal Aspects of Multimedia. In Wolfgang Herzner und Frank Kappe (Hrsg.), *Multimedia/Hypermedia in Open Distributed Environments*. Eurographics Symposium, Graz, Springer, Juni 1994.
- [FLYV92] V. Fuller, T. Li, J. Yu und K. Varadhan. RFC 1338: Supernetting: an Address Assignment and Aggregation Strategy, Juni 1992.
- [For92] Fore Systems, Inc. SPANS: Simple Protocol for ATM Network Signalling, Version 2.1.1. Technischer Bericht, Fore Systems, Inc, Pittsburgh, Pennsylvania, U.S.A., 1992.
- [For93] Fore Systems, Inc. ATM Networks. Technischer Bericht, Fore Systems, Inc, Pittsburgh, Pennsylvania, U.S.A., September 1993.
- [FR95] Oliver Frick und Martin Richartz. Exploiting High Speed Networks for Education. In *Workshop: Interactive and Distributed Multi-Media Systems on Highspeed Networks, on: Third International World-Wide Web Conference*, Darmstadt, april 1995.
- [FRG93] Oliver Frick, Tom Rüdebusch und Hans-W. Gellersen. Some Developers', End-Users', and General Issues in CSCW. In *ECSCW'93 Tools and Technologies Workshop*, Milano, Italy, September 1993.
- [Fri94] Oliver Frick. The Berkomp Multimedia Teleservices. Vortrag auf dem Eurographics Symposium, Graz, Juni 1994.
- [FvD90] J. Foley und A. van Dam. *Fundamentals of Interactive Computer Graphics*. Addison-Wesley, Reading, MA. 1990.
- [FWW⁺93] Tim Finin, Jay Weber, Gio Wiederhold, Michael Genesereth, Richard Fritzson, James McGuire, Stuart Shapiro und Chris Beck. DRAFT Specification of the KQML Agent-Communication Language. The DARPA Knowledge Sharing Initiative, External Interfaces Working Group, Juni 1993.
- [GB92] Les Gasser und Jean-Pierre Briot. *Distributed Artificial Intelligence: Theory and Praxis*, Kapitel Object-Based Concurrent Programming and Distributed Artificial Intelligence, Seite 81–107. Kluwer. 1992.
- [Gel93] Hans-W. Gellersen. Objekte in verteilten Systemen 9. Interner Bericht 24/93, Universität Karlsruhe, Fakultät für Informatik, 1993.
- [Gel94a] Hans-W. Gellersen. Aspects in User Interface Design for Mobile Multi-User Applications. In B. Wolfinger (Hrsg.), *Innovationen bei Rechen- und Kommunikationssystemen*, Informatik Aktuell, Seite 134–141. Springer-Verlag, Hamburg, August 28–September 2 1994. Tagungsband der 24. GI-Jahrestagung.
- [Gel94b] Hans-W. Gellersen. Integration of Concerns in User Interface Development for Mobile Multi-User Applications. In S. Howard und Y. Leung (Hrsg.), *Harmony through working together*, Melbourne, Australia, November 28–Dezember 1 1994. Seite 301–307. OZCHI'94 Conference Proceedings.
- [Gel94c] Hans-W. Gellersen. Multimodale User Interfaces und Telekooperation. Interner Bericht 20/94, Universität Karlsruhe, Fakultät für Informatik, 1994.

- [Gel95] Hans-W. Gellersen. Support of User Interface Design Aspects in a Framework for Distributed Cooperative Applications. In R. Taylor (Hrsg.), *Software Engineering and Human Computer Interaction*, Nr. 896 der Lecture Notes in Computer Science. Springer-Verlag, 1995.
- [Gem90] Prof. Dr. H. Gemünden. Theoretische Ansätze zur organisatorischen Gestaltung. Vorlesung, Institut für Angewandte Betriebswirtschaftslehre und Unternehmensführung, Universität Karlsruhe, 1990.
- [Ger94] W. Gerteis. *Interaktive Führung beim Entwurf komplexer Software-Systeme*. Verlag Shaker. 1994.
- [GHM94] A. Gupta, W. Howe und M. Moran. *Scalable resource reservation for multi-party real-time communication*. Technical Report TR-94-050, Tenet Group, University of California at Berkeley. 1994.
- [GK94] S. Gessler und A. Kotulla. PDAs as mobile WWW-Browsers. In *2nd int. Conference on WWW*, Chicago (USA), October 1994. NCSA.
- [GKK95] A.G. Grosse, L. Keller und D.A. Kottmann. Verteilungstechnologische Grundlagen integrierter Systeme. In *Proc. SFB 346 Workshop Integrierte Planung flexibler Produktionssysteme*, 8. März 1995, Karlsruhe, 1995.
- [GM95a] Hans-W. Gellersen und M. Mühlhäuser. Design of Workplace-integrating User Interfaces based on Work Scenario Graphs. In D. Benyon und P. Palanque (Hrsg.), *Critical Issues in User Interface Systems Engineering*. Springer-Verlag, London, 1995.
- [GM95b] Hans-W. Gellersen und Max Mühlhäuser. Arbeitsplatzintegration und Medienintegration: Mensch-Computer-Interaktion in kooperativen Anwendungen. In Heinz-Dieter Böcker (Hrsg.), *Software-Ergonomie '95: Mensch-Computer-Interaktion, Anwendungsbereiche lernen voneinander*, Nr. 45 der Berichte des German Chapter of the ACM, Darmstadt, Februar20-23 1995. B.G. Teubner Stuttgart, Seite 143-162.
- [GMK95] S. Gessler, M. Mühlhäuser und A. Kotulla. WWW/Newton: Mobile Computing for the Internet. In *ED-MEDIA '95*, Graz (Austria), June 1995. AACE.
- [Gol89] D.E. Goldberg. *Genetic Algorithms in search, optimization and machine learning*. Addison-Wesley. 1989.
- [Gol90] Fred Goldstein. Compatibility of BLINKBLT with the ATM Adaptation Layer. In *Technical report, Digital Equipment Corporation*, ANSI Technical Subcommittee T1S1.5 Working Group Contribution T1S1.5/90-009, Februar 1990.
- [Gol92] Richard A. Golding. Weak Consistency Group Communication for Wide-Area Systems. In *Proc. 2nd Workshop on the management of Replicated Data*. ACM, November 1992, Seite 13-16.
- [Gre83] S. Greif. *Konzepte der Organisationspsychologie*. Hans Huber Verlag, Bern. 1983.
- [Gro94] E. Gros (Hrsg.). *Anwendungsbezogene Arbeits-, Betriebs- und Organisationspsychologie*. Verlag für Angewandte Psychologie, Göttingen. 1994.
- [GSFY94] P. Gomes Soares Florissi und Y. Yemini. Management of Application Quality of Service. In *Workshop on Distributed Systems Operation and Management, DSOM*, Toulouse, France, 1994. IFIP.
- [GW93] P. Grillo und S. Waldbusser. Host Resources MIB. Request for Comments 1514, Network Working Group, September 1993.
- [HA93] H. Hegering und S. Abeck. *Integriertes Netz- und Systemmanagement*. Addison-Wesley. 1993.
- [Hau22] J. Hauschildt. Wider die Gleichmacherei der Organisation von Führungsentscheidungen. Technischer Bericht, Institut für Betriebswirtschaftslehre der Universität Kiel, Nr. 122.
- [Hei93] J. Heinanen. RFC 1483: Multiprotocol Encapsulation over ATM Adaptation Layer 5, Juli 1993.
- [Hin94a] Robert M. Hinden. Internet Draft: IP Next Generation Overview. <ftp://ds.internic.net/internet-drafts/draft-hinden-ipng-overview-00.txt>, Oktober 1994.
- [Hin94b] Robert M. Hinden. RFC 1710: Simple Internet Protocol Plus White Paper, Oktober 1994.
- [Hin95] Robert M. Hinden. Internet Draft: Internet Protocol, Version 6 (IPv6) Specification. <ftp://ds.internic.net/internet-drafts/draft-ietf-ipngwg-ipv6-spec-01.txt>, März 1995.
- [Hof94] Markus Hofmann. *Zuverlässige Gruppenkommunikation in heterogenen Netzen*. Diplomarbeit am Institut für Telematik, Universität Karlsruhe. August 1994.
- [Hog89] Hogrefe. *Estelle, LOTOS und SDL*. Springer Verlag. 1989.
- [HP91] N. C. Hutchinson und L. L. Peterson. The x-kernel: An Architecture for Implementing Network Protocols. *IEEE Transactions on Software Engineering* 17(1), January 1991, Seite 64-76.
- [Hui90] Christian Huitema. MAVROS: Highlights on an ASN.1 Compiler. *interner Bericht*, 1990.
- [Hut89] E. Hutchins. Metaphors for Interface Design. In *The Structure of Multimodal Dialogue*, Nr. 4 der Human Factors in Information Technology. North-Holland, Amsterdam, 1989.
- [IEE87] IEEE. *Standard VHDL Language Reference Manual*, 1987. Std 1076-1987.
- [ISO86] ISO Standard, IS 8072. *Information processing systems — Open Systems Interconnection — Transport Service Definition*, 1986.
- [ISO88] ISO Standard IS 8073. *Information processing systems — Open Systems Interconnection — Transport Protocol Definition*, 1988.
- [ISO89] ISO/IEC. ISO/IEC IS 7498-4: Information Processing Systems – Open Systems Interconnection – Basic Reference Model – Part 4: Management Framework. International Standard, 15. November 1989. Erste Ausgabe.
- [ISO93a] ISO/IEC. WD 10746-1, Reference Model for Open Distributed Processing – Part 1: Overview and Guide to Use. Working Draft, 1993.

- [ISO93b] ISO/IEC. WD 10746-4, Reference Model for Open Distributed Processing – Part 4: Architectural Semantics. Working Draft, 1993.
- [ISO94a] ISO/IEC. WD 10746-2, Reference Model for Open Distributed Processing – Part 2: Descriptive Model. Draft International Standard, February 1994.
- [ISO94b] ISO/IEC. WD 10746-3, Reference Model for Open Distributed Processing – Part 3: Prescriptive Model. Draft International Standard, February 1994.
- [IT93] ITU-T. General Aspects of Quality of Service and Network Performance in Digital Networks, Including ISDN. CCITT I.350, 1993.
- [ITN92] Ito, Takeuchi und Neufeld. Evaluation of a Multiprocessing Approach for OSI Protocol Processing. Juni 1992. First International Conference on Computer Communications and Networks, San Diego, CA.
- [ITU92] ITU-T Draft Recommendation Q.93B. *B-ISDN User-Network Interface Layer 3 Specification for Basic Call/Bearer Control*. Geneva, Switzerland, October 1992.
- [ITU93a] ITU-T Draft Recommendation Q.2120. *B-ISDN meta-signaling protocol connection oriented protocol (SSCOP)*. Geneva, Switzerland, 17. December 1993.
- [ITU93b] ITU-T Draft Recommendation Q.2650. *B-ISDN interworking between SS No. 7 broadband ISDN user part (B-ISUP) and digital subscriber SS No. 2 (DSS 2)*. Geneva, Switzerland, 17. December 1993.
- [ITU93c] ITU-T Draft Recommendation Q.2660. *Broadband integrated services digital network (B-ISDN) – Interworking between SS No. 7 broadband ISDN part (B-ISUP)*. Geneva, Switzerland, 17. December 1993.
- [ITU93d] ITU-T Draft Recommendation Q.2661. *Broadband integrated services digital network (B-ISDN) functional description of B-ISDN user part (B-ISUP) SS No. 7*. Geneva, Switzerland, 17. December 1993.
- [ITU93e] ITU-T Draft Recommendation Q.2662. *Broadband integrated services digital network (B-ISDN) — General functions messages, signals of B-ISDN user part (B-ISUP)*. Geneva, Switzerland, 17. December 1993.
- [ITU93f] ITU-T Draft Recommendation Q.2663. *Broadband integrated services digital network (B-ISDN) — SS No. 7 B-ISUP user part B-ISUP — Formats and codes*. Geneva, Switzerland, 17. December 1993.
- [ITU93g] ITU-T Draft Recommendation Q.2664. *Broadband integrated service digital network (B-ISDN) SS No. 7 B-ISDN user part (B-ISUP) Basic call procedures*. Geneva, Switzerland, 17. December 1993.
- [ITU93h] ITU-T Recommendation I.121. *Broadband aspects of ISDN (B-ISDN Protocol Reference Model)*. Geneva, Switzerland, 16. June 1993.
- [ITU94a] ITU-T Draft Recommendation Q.1400. *Architecture framework for the development of signalling and OAM protocols using OSI concepts*. Geneva, Switzerland, July 1994.
- [ITU94b] ITU-T Draft Recommendation Q.2100. *B-ISDN signalling ATM adaptation layer (SAAL) overview description*. Geneva, Switzerland, 29. July 1994.
- [ITU94c] ITU-T Draft Recommendation Q.2110. *B-ISDN ATM adaptation layer — Service specific connection oriented protocol (SSCOP)*. Geneva, Switzerland, 29. July 1994.
- [ITU94d] ITU-T Draft Recommendation Q.2130. *B-ISDN signalling ATM Adaptation Layer — Service specific coordination function for support of signalling at the user-network interface (SSCF at UNI)*. Geneva, Switzerland, 29. July 1994.
- [ITU94e] ITU-T Draft Recommendation Q.2931. *Edinburgh TD 155, Broadband Integrated Services Digital Network (B-ISDN), Digital Subscriber Signalling System No.2, User Network Interface Layer 3 Specification for Basic Call/Connection Control*. Geneva, Switzerland, 13. - 21. June 1994.
- [ITU94f] ITU-T Draft Recommendation Q.2961. *Draft Text for Q.2961, Negotiation/Renegotiation: Traffic and QoS Parameters*. Geneva, Switzerland, 1994.
- [ITU94g] ITU-T Draft Recommendation Q.2962. *Draft Text for Q.2962, Negotiation/Renegotiation: Traffic and QoS Negotiation during Call Establishment*. Geneva, Switzerland, 1994.
- [ITU94h] ITU-T Draft Recommendation Q.2963. *Draft Text for Q.2963, Negotiation/Renegotiation: Traffic and QoS Negotiation during Active Phase*. Geneva, Switzerland, 1994.
- [ITU94i] ITU-T Draft Recommendation Q.298x. *Draft Text for Q.298x, Edinburgh TD 152 Rev 1 & 147, Broadband Integrated Services Digital Network (B-ISDN), Digital Subscriber Signalling System No. 2, User Network Interface Layer 3 Specification for Point-Point Multiconnection Call Control*. Geneva, Switzerland, 13.-21. June 1994.
- [ITU94j] ITU-T Recommendation I.363. *B-ISDN ATM Adaptation Layer (AAL) specification*. Geneva, Switzerland, 4. August 1994.
- [ITU94k] ITU-T Recommendation I.371. *Traffic Control and Congestion Control in B-ISDN*. Geneva, Switzerland, 30. March 1994.
- [ITU94l] ITU-T Recommendation Q.700. *Introduction to CCITT Signalling System No. 7*. Geneva, Switzerland, 1. June 1994.
- [ITU94m] ITU-T Recommendation Q.701. *Functional description of the message transfer part (MTP) of signalling system no. 7*. Geneva, Switzerland, 27. April 1994.
- [ITU94n] ITU-T Recommendation Q.761. *Functional description of the ISDN user part of signalling system no. 7*. Geneva, Switzerland, 27. April 1994.

- [ITU94o] ITU-T Recommendation Q.762. *General function of messages and signals of the ISDN user part of signalling system no. 7*. Geneva, Switzerland, 27. April 1994.
- [ITU94p] ITU-T Recommendation Q.763. *Formats and codes of the ISDN user part signalling procedures*. Geneva, Switzerland, 27. April 1994.
- [ITU94q] ITU-T Recommendation Q.764. *Signalling System No. 7 — ISDN user part signalling procedures*. Geneva, Switzerland, 1. Juni 1994.
- [ITU94r] ITU-T Recommendation Q.930. *Digital subscriber signalling system no. 1 (DSS 1) — ISDN user-network interface layer 3 — General aspects*. Geneva, Switzerland, 12. August 1994.
- [ITU94s] ITU-T Recommendation Q.931. *Digital subscriber signalling system no. 1 (DSS 1) — ISDN user-network interface layer 3 specification for basic call control*. Geneva, Switzerland, 22. November 1994.
- [Jab95] Stephan Jablonski. Workflow-Management-Systeme: Motivation, Modellierung, Architektur. *Informatik Spektrum* 18(1), 1995, Seite pp. 13–24.
- [Jan94] M. Jander. Management Frameworks - Moving Toward a Unified View of Distributed Networks. *Data Communications International* 23(3), Februar 1994, Seite 58 – 68. McGraw-Hill's Network Technology Magazine.
- [Jef94] Ron Jeffries. ATM LAN Emulation: The Inside Story. *Data Communications*, Sep. 21 1994, Seite 95–100.
- [KEG93] König, Effelsberg und Gotzhein. Ableitung parallel ausführbarer Protokollimplementierungen. *PIK*, 1993.
- [Kel93] L. Keller. Vom Name-Server zum Trader — Ein Überblick über Trading in verteilten Systemen. *Praxis der Informationsverarbeitung und Kommunikation (PIK)* 16(3), September 1993, Seite 122–133.
- [Kel94] L. Keller. Trading of Complex Services in Distributed Systems. In *5th IFIP/IEEE International Workshop on Distributed Systems: Operations & Management (DSOM'94)*, Toulouse, Frankreich, Oktober 1994.
- [Ker94] H.M. Kern. Verteilungsanforderungen in Maschinenbauanwendungen. Diplomarbeit, Diplomarbeit, Fakultät für Informatik, Institut für Telematik, Universität Karlsruhe, 1994.
- [KG94] Dietmar A. Kottmann und Stefan Gessler. Mobile Datenkommunikation und ihre Auswirkungen auf die Anwendungsunterstützung. *HMD — Theorie und Praxis der Wirtschaftsinformatik, Heft 177*, Mai 1994, Seite 84–97.
- [KG95] L. Keller und A.G. Grosse. Vermittlung zuverlässiger Dienste in Trading-basierten Systemen. In *Workshop Anwendungsunterstützung für heterogene Rechnernetze, März 1995, Freiberg/Sachsen*, 1995.
- [KK94a] L. Keller und D.A. Kottmann. Verwaltung lose-gekoppelter Datenbankcluster mittels Trading. In *Proc. 39. Internationales Wissenschaftliches Kolloquium, Ilmenau, S. 194-199*, 1994.
- [KK94b] D.A. Kottmann und L. Keller. Integrating Distribution and Mobility into an Object-Oriented Database. In *Patel D., Sun Y., Patel S.: Proc. International Conference on Object Oriented Information Systems (OOIS'94)*, Springer Verlag, London, Dezember, S. 350-363, 1994.
- [KK94c] D.A. Kottmann und L. Keller. Transparent Support for Object Mobility in an Object-Oriented Database. In *Proc. International Symposium on Applied Corporate Computing (ISACC'94)*, Monterrey, Mexiko, Oktober, S. 85-94, 1994.
- [KK95] L. Keller und D.A. Kottmann. Verteilung und Mobilität in einer objektorientierten Datenbank. In *Franke K., Hübner U., Kalfa W.: Proc. Kommunikation in verteilten Systemen (KIVS'95)*, Informatik Fachberichte, Springer Verlag, S. 285-299, 1995.
- [KKK⁺94] L. Keller, C. Kilger, D.A. Kottmann, G. Moerkotte, A.B. Schill, H.-D. Walter und A. Zachmann. Aktive und mobile Objekte als Modellierungskonzept für dezentrale Ingenieur Anwendungen. In *3. GI/ITG/GMA Fachtagung Softwaretechnik in Automation und Konstruktion — Datenbanken unter Realzeit- und Entwicklungsanforderungen (STAK'94)*, Ilmenau, S. 75-100, März, 1994.
- [KKS93] Krishnakumar, Kneuer und Shaw. HIPOD: An Architecture for High-Speed Protocol Implementations. 1993, Seite 383–396.
- [KLW95] D.A. Kottmann, P.C. Lockemann und H.-D. Walter. Multi-Object Cooperation in Distributed Object Bases. Technischer Bericht, 16/95, Fakultät für Informatik, Universität Karlsruhe, 1995.
- [KO90] E.S. Kuh und T. Ohtsuki. Recent Advances in VLSI Layout. In *Proceedings of the IEEE*, Februar 1990, Seite 237–263.
- [Kot93] Dietmar A. Kottmann. Unterstützung der Datenhaltung in mobilen Systemen. In *Entwicklung und Management verteilter Anwendungssysteme*, Frankfurt am Main, Oktober 1993. GI/ITG Fachgruppe Kommunikation und verteilte Systeme.
- [Kot95a] Dietmar A. Kottmann. Abgekoppelte Objekte in heterogenen Umgebungen mit mobilen Teilnehmern. In *Workshop Anwendungsunterstützung für heterogene Rechnernetze*, Freiberg/Sachsen, März 1995.
- [Kot95b] Dietmar A. Kottmann. Datenmanagement im Mobile Computing. *HMD — Theorie und Praxis der Wirtschaftsinformatik*, Erscheint in Heft 184, Juli 1995.
- [KP88] G.E. Krasner und S.T. Pope. A Cookbook for using the Model-View-Controller User Interface Paradigm in Smalltalk-80. *Journal of Object-Oriented Programming*, August 1988.

- [KS92] James J. Kistler und M. Satyanarayanan. Disconnected Operations in the CODA File System. *ACM Transactions on Computer Systems* 10(1), February 1992, Seite 3–25.
- [KS95a] D.A. Kottmann und A.B. Schill. Flexible and Transparent Fault-Tolerance for Distributed Object-Oriented Applications. Technischer Bericht, 20/95, Fakultät für Informatik, Universität Karlsruhe, März 1995.
- [KS95b] R. Kraut und L. Streeter. Coordination in Software Development. *Communications of the ACM* 38(3), März 1995, Seite 69–81.
- [KSCK93] Henry A. Kautz, Bart Selman, Michael Coen und Stephen Ketchpel. An Experiment in the Design of Software Agents. Technischer Bericht, AI Principles Research Department, AT&T Bell Laboratories, Murray Hill, 1993.
- [KTZ94a] O. G. Koufopavlou, A. N. Tantawy und M. Zitterbart. A Comparison of Gigabit Router Architectures. Conference on High Performance Networking, Juni 1994.
- [KTZ94b] O. G. Koufopavlou, A. N. Tantawy und M. Zitterbart. IP-Routing among Gigabit Networks. Second International Symposium on Interworking, Mai 1994.
- [KW94] E. Kovacs und S. Wirag. Trading and Distributed Application Management. In *Workshop on Distributed Systems Operation and Management, DSOM*, Toulouse, France, 1994. IFIP.
- [Lau94] M. Laubach. RFC 1577: Classical IP and ARP over ATM, Januar 1994.
- [LO94] Leue und Oechslin. From SDL Specifications to Optimized Parallel Protocol Implementations. IFIP, 1994, Seite 309–327. 4th International Workshop on Protocols for High-Speed Networks, Vancouver.
- [LR93] Daniel C. Lynch und Marshall T. Rose (Hrsg.). *Internet System Handbook*. Addison-Wesley. 1993.
- [Lut95] Lutz. Einsatz von Model Checking bei der Verifikation von hochleistungsfähigen Kommunikationssystemen. Diplomarbeit, Universität Karlsruhe, Institut für Telematik, Februar 1995.
- [M. 94] M. Walch. *A Formal Specification of the TEMPO Protocol*. Berlin, Germany, 1. March 1994.
- [Mae94] P. Maes. Agents that reduce work and information overload. *Communications of the ACM* 37(7), 1994, Seite 31–40.
- [Mas94] John Massey. SAFER K-64: A Byte-Oriented Block-Ciphering Algorithm. In *Lecture Notes in Computer Science 809*, Berlin, Heidelberg, 1994. Seite 1–17.
- [MB94] L. Mathy und O. Bonaventure. *QoS Negotiation for Multicast Communications*. Proceedings of International COST 237 Workshop, Vienna. 1994.
- [McA93] McAuley. Operating System Support for the Desk Area Network. Nr. 846 der LNCS. Springer Verlag, 1993.
- [McM93] McMillan. *Symbolic Model Checking*. Kluwer Academic Publishers. 1993.
- [MG95] Max Mühlhäuser, Hans-Werner Gellersen und Stefan Gessler. Distributed Client Architectures for advanced WWW applications. In *Interactive and Distributed Multi-Media Systems on Highspeed Network*, Darmstadt (Germany), April 1995.
- [Mic92] Z. Michalewicz. *Genetic algorithms + data structures = evolution programs*. Springer-Verlag. 1992.
- [Min91] S. E. Minzer. A Signalling Protocol for Complex Multimedia Services. *IEEE Journal on Selected Areas in Communications* 9(9), December 1991, Seite 1383–1394.
- [MM94] M. Mock und A. Müller. Generation of Distributed Services with Diogenes. In M. Medina und N. S. Borenstein (Hrsg.), *IFIP International Conference on Upper Layer Protocols, Architecture and Applications*, Seite 13 – 28. Spain, Juni 1994.
- [MMC92] BerKom Working Group MMC. *The BerKom Multimedia Collaboration Teleservice - Release 2.0*. BerKom Technical Report. 1992.
- [MMC93] MMC. The Berkom Multimedia Collaboration Teleservice. Grobspezifikation 3.2, DeTeBerkom GmbH, Oktober 1993.
- [MMT94] BerKom Working Group MMT. *The BERKOM-II MultiMedia Transport System (MMT), Version 5.0*. BerKom Technical Report. 1994.
- [MSS93] Eckard Moeller, Angela Scheller und Gerd Schürmann. Der BERKOM-Teledienst Multimedia Mail. *PIK* 16(3), 1993, Seite 134–139.
- [MT92a] P. A. Miller und P. N. Turcu. Generic Signalling Protocol: Architecture, Model, and Services. *IEEE Transactions on Communications* 40(5), May 1992, Seite 957–966.
- [MT92b] P. A. Miller und P. N. Turcu. Generic Signalling Protocol: Switching Networking, and Internetworking. *IEEE Transactions on Communications* 40(5), May 1992, Seite 967–979.
- [MU94] M. McGovern und R. Ullmann. RFC 1707: CATNIP: Common Architecture for the Internet, Oktober 1994.
- [Net93] Netcomm, Ltd. DV 2 — Switch Installation Managers Guide, Version 2.0. Technischer Bericht, Netcomm, Ltd., Basildon, Essex, U.K., 1993.
- [Neu75] K. Neumann. *Operations Research Verfahren Band III*. Carl Hanser Verlag. 1975.
- [Obj93a] Object Design, Inc., 25 Burlington Mall Road, Burlington, MA 01803. *ObjectStore – Plattform Guide*, Ausgabe ObjectStore Release 3 for ULTRIX, December 1993.
- [Obj93b] Object Design, Inc., 25 Burlington Mall Road, Burlington, MA 01803. *ObjectStore – User Guide: LI*, Ausgabe ObjectStore Release 3 for ULTRIX, December 1993.
- [Pfa85] G.E. Pfaff (Hrsg.). *User Interface Management Systems*. Springer-Verlag, Berlin. 1985.

- [PLM⁺94] M. Perez, F. Liaw, A. Mankin, E. Hoffman, D. Grossman und A. Malis. Internet Draft: ATM Signaling Support for IP over ATM. <ftp://ds.internic.net/internet-drafts/draft-ietf-ipatm-sig-02.txt>, 1994.
- [Pow91] R. J. D. Power (Hrsg.). *Cooperation Among Organizations*, Band 1 der *Research Reports ESPRIT: Project 5660 PECOS*. Springer Verlag. 1991.
- [PT87] C. Partridge und Glenn Trewit. The High-Level Entity Management System. Request for Comments 1021 – 1024, October 1987. DDN Network Information Center, SRI International.
- [PZ92] Pei und Zukowski. A Compiler for Communication Network Interface Circuits. IEEE, 1992, Seite 1293–1297.
- [R. 94] R. Händel and M. Huber and S. Schröder. *ATM Networks: Concepts, Protocols, Applications*. Addison-Wesley, Reading, Massachusetts, U.S.A. 1994.
- [RB92] Ch. Brown R. Bianchini. Parallel Genetic Algorithms on Distributed-Memory Architectures. Technischer Bericht, The University of Rochester, August 1992.
- [RMDB92] L. Roberts, T. MacDonald, K. Duffie und G. Bernstein. Fast Select Virtual Circuit Routing for B-ISDN Network. Technischer Bericht, NetExpress, Foster City, California, U.S.A., 1992.
- [Rö95] J. Röthig. *Kooperatives, adaptives Ressourcen-Management in ATM-Netzwerken*. VDI Verlag. 1995.
- [Ros86] F.E. Ross. FDDI — A Tutorial. *IEEE Communications Magazine* 24(5), May 1986, Seite 10–17.
- [RSFW94] A. Richards, A. Seneviratne, M. Fry und V. Witana. Tailoring the Transport Protocol for Giga Bit Networks. In *Australian Telecommunication Networks and Applications Conference*, Dezember 1994.
- [Rya93] B. Ryan. Communications Get Personal. *Byte*, February 1993, Seite 169–176.
- [SB93] Jochen Schiller und Torsten Braun. VLSI-Implementation Architecture for Parallel Transport Protocols. In *Workshop on VLSI in Communications*, Lake Tahoe, CA, September 1993. IEEE, Seite 76–77.
- [SCB92] S. Sheng, A. Chandrakasan und R. Broderon. A Portable Multimedia Terminal. *IEEE Communications Magazine* 30(12), December 1992, Seite 64–76.
- [Sch94a] Jochen Schiller. VLSI-Implementierungsarchitektur für parallele Transportprotokolle. In *1. Arbeitstreffen Entwurf und Implementierung von Hochleistungs-Kommunikationssystemen*, Karlsruhe, Januar 1994. GI/ITG.
- [Sch94b] E. Schöneburg. *Genetische Algorithmen und Evolutionsstrategien*. Addison-Wesley. 1994.
- [SDW92] Strayer, Dempsey und Weaver. *XTP: The Xpress Transfer Protocol*. Addison-Wesley. 1992.
- [Sea69] J. R. Searle. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press. 1969.
- [Sei94a] J. Seitz. *Integration heterogener Netzwerkmanagementarchitekturen*. Nr. 289 der Fortschrittsberichte VDI, Reihe 10: Informatik/Kommunikationstechnik. VDI-Verlag, Düsseldorf. 1994.
- [Sei94b] J. Seitz. *Netzwerkmanagement*. Nr. 2 der Thomsons's Aktuelle Tutorien (TAT). International Thomson Publishing, Bonn. Ausgabe 1, 1994.
- [Sei94c] J. Seitz. OSF DCE und DME: Neue Industriestandards für Rechnernetze und verteilte Anwendungen — Teil 7: Eine Übersicht über das OSF-Distributed Management Environment. *unix/mail* 12(3), Juni 1994, Seite 179 – 184.
- [Sei94d] J. Seitz. OSF DCE und DME: Neue Industriestandards für Rechnernetze und verteilte Anwendungen — Teil 8: Das „Traditional Management Framework“ von OSF/DME. *unix/mail* 12(4), August 1994, Seite 377 – 383.
- [Sei94e] J. Seitz. OSF DCE und DME: Neue Industriestandards für Rechnernetze und verteilte Anwendungen — Teil 9: Das „Object Oriented Framework“ von OSF/DME. *unix/mail* 12(5), Oktober 1994, Seite 432 – 436.
- [Sei94f] J. Seitz. OSF DCE und DME: Neue Industriestandards für Rechnernetze und verteilte Anwendungen — Teil 10: Managementanwendungen in OSF/DME. *unix/mail* 12(6), Dezember 1994, Seite 513 – 517.
- [Sei94g] J. Seitz. Towards Integrated Network Management: A Generic Management Protocol. In *Fifth IFIP/IEEE International Workshop on Distributed Systems: Operations & Management DSOM'94*, 118, Route de Narbonne, 31062 Toulouse Cedex (France), 10. – 12. Oktober 1994. IFIP Working Group 6.6 Network Management; IEEE CNOM; Centre Interuniversitaire de Calcul de Toulouse; Conseil Régional Midi-Pyrénées; Hewlett Packard; Institut de Recherche en Informatique de Toulouse; Université Paul Sabatier, Université Paul Sabatier – IRI/SIERA, Seite 6–1 — 6–17.
- [Sei95] J. Seitz. OSF DCE und DME: Neue Industriestandards für Rechnernetze und verteilte Anwendungen — Teil 11: OSF/DME — Produkte und Zukunftsaussichten. *unix/mail* 13(1), Februar 1995, Seite 45 – 50.
- [Shi91] D.R. Shier. *Network Reliability and Algebraic Structures*. Clarendon Press, Oxford. 1991.
- [Sho90] Y. Shoham. Agent Oriented Programming. Technischer Bericht Stan-Cs-90-1335, Stanford University, Palo Alto, CA, 1990.
- [Shu88] Nan C. Shu. *Visual Programming*. Van Nostrand Reinhold. 1988.
- [SJ93] Steiger und Jostema. PAR-SDL: Software design and implementation for transputer systems. 1993. Transputer Applications and Systems '93.

- [SKM⁺93] M. Satyanarayanan, James J. Kistler, Lily B. Mummert, Maria R. Ebling, Puneet Kumar und Qi Lu. Experience with Disconnected Operations in a Mobile Computing Environment. In *Proc. Symposium on Mobile & Location Independent Computing Systems*. Usenix, August 1993, Seite 11–28.
- [SKO⁺94] Robert J. Souza, P.G. Krishnakumar, Cüneyt M. Özveren, Robert J. Simcoe, Barry A. Spinney, Robert E. Thomas und Robert J. Walsh. The GIGAswitch System: A High-Performance Packet Switching Platform. Digital Technical Journal 6, Digital Equipment Corporation, 1994.
- [SNR90] Sabnani, Netravali und Roome. Design and Implementation of a High-Speed Transport Protocol. *IEEE Transactions on Communications* 38(11), November 1990, Seite 2010–2024.
- [Sol94] G. Solvie. *Adaptive Kommunikationsunterstützung in offenen Systemen*. VDI Verlag. 1994.
- [SP91] Sterbenz und Parulkar. AXON Host-Network Interface Architecture for Gigabit Communications. North-Holland, 1991, Seite 211–236. Protocols for High-Speed Networks II.
- [SSSZ94a] D. C. Schmidt, B. Stiller, T. Suda und M. Zitterbart. Configuring Function-based Communication Protocols for Multimedia Applications. IFIP Conference on Upper Layer Protocols, Architectures and Applications, Juni 1994.
- [SSSZ94b] Douglas C. Schmidt, Burkhard Stiller, Tatsuya Suda und Martina Zitterbart. Configuring Function-based Communication Protocols for Multimedia Applications. In *IFIP 6.5 International Working Conference on Upper Layer Protocols, Architectures, and Applications, ULPAA '94*, Amsterdam, Netherlands, 1. – 3. June 1994. IFIP Transactions C-14, North Holland, Seite 361–367.
- [SSZ94] Claudia Schmidt, Burkhard Stiller und Martina Zitterbart. Towards Flexible Service-integrated Communication Subsystems. In *IASTED/ISMM International Conference on Distributed Multimedia Systems and Applications*, Honolulu, Hawaii, U.S.A., 15.-17. August 1994. Seite 361–367.
- [Sta87] W. H. Staehle. *Management, eine verhaltenswissenschaftliche Einführung*. München. Ausgabe 4, 1987.
- [Ste94] Steenkiste. A Systematic Approach to Host Interface Design for High-Speed Networks. *IEEE Computer*, 1994.
- [Sti94a] B. Stiller. CODCOM: A Code Configuration Manager in the Function-based Communication Subsystem. In *4th Open Workshop on High Speed Networks*, Brest, France, 7.-9. September 1994.
- [Sti94b] B. Stiller. *Flexibler Protokollkonfiguration zur Unterstützung eines diensteintegrierenden Kommunikationssubsystems*, Band 10, no. 306. VDI, Düsseldorf, Germany. 16. February 1994.
- [Sti94c] B. Stiller. FuKSS: Ein funktionsbasiertes Kommunikationssystem zur flexiblen Konfiguration von Kommunikationsprotokollen. In *1. GI/ITG Arbeitstreffen zur Architektur und Implementierung von Hochleistungs-Kommunikationssystemen*, Karlsruhe, Germany, 17.-18. January 1994. Seite Session 3, Paper 1.
- [Sti94d] B. Stiller. PROCOM: A Protocol Configuration Manager in the Function-based Communication Subsystem. In *First International Workshop on High Performance Protocol Architectures*, Sophia-Antipolis, France, 15.-16. December 1994. Seite Session 3, Paper 9.
- [STZ94] D. N. Serpanos, A. N. Tantawy und M. Zitterbart. A High Performance Transparent Bridge. *ACM/IEEE Transactions in Networking* 2(4), August 1994, Seite 352 – 362.
- [Sug81] K. Sugiyama. Methods for visual understanding of hierarchical systems. In *IEEE Transactions on Systems*, 1981.
- [Svo89] L. Svobodova. Measured Performance of Transport Service in LANs. In *Computer Networks and ISDN Systems*, Band 18, 1989, Seite 31–45.
- [Syn94] Synopsys Inc. *User Manuals*, 1994. Mountain View.
- [Syr93] Michael Syring. CSCW (Computer Supported Cooperative Work) in Organisationen - Grundlagen und Probleme. In Stefan Kirn et al. (Hrsg.), *CSCW im Spannungsfeld von Dezentralisierung und Integration*, Giessen, sept 1993.
- [SZ94] Claudia Schmidt und Martina Zitterbart. Towards Integrated QoS Management. In *First International Workshop on High Performance Protocol Architectures HIPPARCH '94*, Sophia Antipolis, Frankreich, Dezember 1994.
- [SZ95a] Jochen Schiller und Martina Zitterbart. Modular VLSI Implementation Architecture for High-Performance Communication Support. In *7th IEEE Workshop on Local and Metropolitan Area Networks*, Marathon, FL, März 1995. IEEE.
- [SZ95b] Claudia Schmidt und Martina Zitterbart. *Internetworking - Brücken, Router und Co.* Nr. 8 der Thomson Aktuelle Tutorien (TAT). International Thomson Publishing, Bonn. 1995.
- [SZ95c] Claudia Schmidt und Martina Zitterbart. QoS Management for Service Integrated Communication Systems. In *7th IEEE Workshop on Local and Metropolitan Area Networks*, Marathon, Florida, U.S.A., März 1995.
- [SZ95d] Claudia Schmidt und Martina Zitterbart. Reservierung von Netzwerk-Ressourcen - Ein Überblick über Protokolle und Mechanismen. *PIK - Praxis der Informationsverarbeitung und Kommunikation*, 1995.
- [Tex94] Texas Instruments Inc. *TMS320C80 Multimedia Video Processor (MVP)*, Juni 1994. Technical Brief.
- [Top90] C. Topolcic. *Experimental Internet Stream Protocol, Version 2 (ST-II)*. RFC 1190. 1990.
- [TZ93a] A. N. Tantawy und M. Zitterbart. An Enhanced Transport Service Interface for Emerging Applications. Symposium on Global Networking, Dezember 1993.

- [TZ93b] A. N. Tantawy und M. Zitterbart. A Scheme for High Performance LAN Interconnection across MANs. *IEEE Journal on Selected Areas in Communications* 11(8), Oktober 1993, Seite 1133 – 1144.
- [Ucl92] *A Guide To Implementing Managed Objects Using The GMS*. Universal College London, Ausgabe 2.99.1, Oktober 1992.
- [Ull93] R. Ullmann. RFC 1475: TP/IX: The Next Internet, Juni 1993.
- [VL87] Varghese und Lauck. Hashed and Hierarchical Timing Wheels: Data Structures for the Efficient Implementation of a Timer Facility. ACM, 1987, Seite 25–38.
- [vM92] F. von Martial. Einführung in die Verteilte Künstliche Intelligenz. *Künstliche Intelligenz: Forschung, Entwicklung, Erfahrungen*. 6(1), März 1992, Seite 6–11. Organ des Fachbereichs Künstliche Intelligenz der GI e.V.
- [vR92] L. von Rosenstiel. *Grundlagen der Organisationspsychologie*. Sammlung Poeschel. Schäffer-Poeschel Verlag Stuttgart. Ausgabe 3, 1992.
- [Wei87] A. B. Weinert. *Lehrbuch der Organisationspsychologie*. Psychologie Verlags Union, München-Weinheim. Ausgabe 2, 1987.
- [Wei91] Mark Weiser. The Computer for the 21st Century. *Scientific American*, September 1991, Seite 66–75.
- [Wei93] Mark Weiser. Some Computer Science Issues in Ubiquitous Computing. *Communications of the ACM* 36(7), July 1993, Seite 74–84.
- [WF86] T. Winograd und F. Flores. *Understanding computers and cognition: A new foundation for design*. Ablex, Norwood, New Jersey. 1986.
- [WHD94] L. C. Wolf, R. G. Herrtwich und L. Delgrossi. *Filtering Multimedia Data in Reservation-Based Internetworks*. IBM European Networking Center, Heidelberg, Germany. 1994.
- [WJ95] Michael J. Wooldridge und Nicholas R. Jennings. *Intelligent Agents - Theories, Architectures, and Languages*, Band 890 der *Lecture Notes in Artificial Intelligence*, Kapitel Agent Theories, Architectures, and Languages: A Survey. Springer Verlag. Januar 1995.
- [WO95] Robert J. Walsh und Cüneyt M. Özveren. The GIGAswitch Control Processor. *IEEE Network* 9(1), Januar/Februar 1995, Seite 36–43.
- [WOB93] Wenban, O’Leary und Braun. Codesign of Communication Protocols. *IEEE Computer*, 1993.
- [WWW] The World Wide Web, a global information initiative. <ftp://ftp.w3.ch/pub/www/doc>.
- [WZ95] Hajo R. Wiltfang und Martina Zitterbart. IP-Prozessor für Gigabit-Router. In K. Franke, U. Hübner und W. Kalfa (Hrsg.), *Kommunikation in Verteilten Systemen*, TU Chemnitz-Zwickau, 22. - 24. Februar 1995. GI/ITG–Fachtagung, Springer-Verlag, Berlin Heidelberg, Seite 530–544.
- [ZDE⁺] Lixia Zhang, Steve Deering, Deborah Estrin, Scott Shenker und Daniel Zappala. RSVP: A New Resource ReSerVation Protocol. Preliminary Draft.
- [Zei94] C. Zeidler. *Reflexive Konfigurationsverwaltung in einer objektorientierten verteilten Umgebung*. Shaker. 1994.
- [Zit93a] M. Zitterbart. Dienstintegrierende Hochleistungsnetze und -systeme für die Breitbandübertragung. Hochschul Computer Forum, Berlin, Oktober 1993.
- [Zit93b] M. Zitterbart. High Performance Transport Protocols and Services. Tutorial bei der 5th International Conference on Data Communication Systems and their Performance, Oktober 1993.
- [Zit93c] M. Zitterbart. Hochleistungskommunikation: Eine Herausforderung. *Praxis der Informationsverarbeitung und Kommunikation* 16(4), Dezember 1993, Seite 192.
- [Zit94a] M. Zitterbart. *High Performance Communications*, Kapitel Parallelism in Communication Subsystems, Seite 177 – 194. Kluwer Academic Publishers. 1994.
- [Zit94b] M. Zitterbart. *Hochleistungskommunikation; Band 1: Technologie und Netze*. Oldenbourg Verlag. 1994.
- [Zit94c] M. Zitterbart. Protokolle für die Hochleistungskommunikation. HMD-Heft 177: Hochgeschwindigkeitsnetze, 1994.
- [Zit94d] M. Zitterbart. Protokolle und Dienste für Hochleistungs-Kommunikationssysteme. 3. Workshop über wissenschaftliches Rechnen: Praxisorientierte Parallelverarbeitung, Oktober 1994.
- [Zit95] M. Zitterbart. *Flexible und effiziente Kommunikationssysteme für Hochleistungsnetze*. Thomson Publishing. 1995.
- [ZS93] C. Zeidler und J. Seitz. Integrational Configuration Management. In *Second Workshop on Networked Systems Management*, Aachen, Germany, 29. – 30. April 1993. Philips Research Laboratories. Folienkopien.
- [ZST93] M. Zitterbart, B. Stiller und A. Tantawy. A Model for High-Performance Communication Subsystems. *IEEE Journal on Selected Areas in Communication* 11(4), May 1993, Seite 507–518.

Veröffentlichungen des Institutes

- [AKTZ94] J. M. Abler, O. G. Koufopavlou, A. N. Tantawy und M. Zitterbart. On the Design of a Multigigabit Router. *Journal on High Speed networks* 3(3), 1994, Seite 209 – 232.
- [BD95] Torsten Braun und Christophe Diot. Protocol Implementation Using Integrated Layer Processing. ACM Sigcomm '95, Cambridge, Massachusetts (USA), 30. August - 1. September, 1995.
- [BDH95] Torsten Braun, Christophe Diot und Anna Hoglander. An Experimental User-Level Implementation of TCP. *interner Bericht*, 1995.
- [Bec93] J. Becher. *Konfigurierung verteilter, heterogener Informationsverarbeitungssysteme*. Verlag Shaker. 1993.
- [Bra94a] Torsten Braun. Architektur, Implementierung und Leistungsbewertung eines parallelen Transportsystems für zellenbasierte Hochgeschwindigkeitsnetze. In *GI/ITG-Fachgruppe 'Kommunikation und Verteilte Systeme': 1. Arbeitstreffen 'Architektur und Implementierung von Hochleistungs-Kommunikationssystemen'*, Karlsruhe, 17./18. Januar 1994.
- [Bra94b] Torsten Braun. PATROCLOS: A Flexible and High-Performance Transport Subsystem. In *4th IFIP Workshop on Protocols for High-Speed Networks*, Vancouver, 10.-12. August 1994. Seite 183–201.
- [Bra95a] Torsten Braun. Ein paralleles Transportsystem für zellenbasierte Hochgeschwindigkeitsnetze. In *Kommunikation in Verteilten Systemen*, Informatik aktuell, Chemnitz, 22.24. Februar 1995. Springer-Verlag, Seite 562–566.
- [Bra95b] Torsten Braun. Limitations and Experiences of Integrated Layer Processing. In *1. Gemeinsame Jahrestagung der Gesellschaft für Informatik (GI) und der Schweizer Gesellschaft für Informatik (SI)*, Zürich, 18.-20. September 1995.
- [BS94] Torsten Braun und Claudia Schmidt. A Parallel Transport Subsystem Implementation for High Performance Communication. *Concurrency: Practice and Experience, Special Issue on High Performance Distributed Computing*, Juni 1994, Seite 375–391.
- [BSSZ94] Torsten Braun, Claudia Schmidt, Burkhard Stiller und Martina Zitterbart. A Protocol Architecture for a Flexible High Performance Communication Subsystem. In *CaberNet Workshop*, Dublin, 23.-25. März 1994.
- [BSZ94a] Torsten Braun, Jochen Schiller und Martina Zitterbart. Implementation of Transport Protocols Using Parallelism and VLSI Components. In *International Telecommunication Symposium 1994 (ITS'94)*, Rio de Janeiro, 22.-25. August 1994. Seite 285–290.
- [BSZ94b] Torsten Braun, Jochen Schiller und Martina Zitterbart. A Modular VLSI Implementation Architecture for Communication Subsystems. In *4th IFIP Workshop on Protocols for High-Speed Networks*, Vancouver, 10.-12. August 1994. Seite 119–134.
- [BZ94] Torsten Braun und Martina Zitterbart. Videokonferenzen im Internet. In *Praxis der Informationsverarbeitung und Kommunikation (PIK)* 17, 4/1994, 1994, Seite 253–254.
- [Car94a] Georg Carle. Adaptation Layer and Group Communication Server for Reliable Multipoint Services in ATM Networks. In *R. Steinmetz (Ed.): Advanced Teleservices and High Speed Communication Architectures; Lecture Notes in Computer Science 882*, Springer-Verlag, S. 124-138, Second International Workshop, IWACA '94, Heidelberg, September 1994.
- [Car94b] Georg Carle. Einsatz von Übertragungswiederholung und Vorwärtsfehlerkorrektur für zuverlässige Multicastkommunikation in ATM-Netzen. In *ITG Fachtagung Codierung für Quelle, Kanal und Übertragung*, München, Oktober 1994.
- [Car94c] Georg Carle. Error Control for Reliable Multipoint Communication in ATM Networks. In *Third International Conference on Computer Communications and Networks ICCCN'94*, S. 33-37, San Francisco, CA, U.S.A., September 1994.
- [Car94d] Georg Carle. Reliable Group Communication in ATM Networks. In *Twelfth Annual European Fibre Optics Communications and Networking Conference EFOCN'94*, S. 30-34, Heidelberg, Juni 1994.
- [Car95] Georg Carle. Provision of Reliable Multipoint Services in ATM Networks. In *International Conference on ATM Developments '95*, Rennes, Frankreich, März 1995.
- [CS95] Georg Carle und Jochen Schiller. Modular VLSI Implementation Architecture for the Provision of High-Performance Multipoint Multimedia Services. In *Silicon Valley Networking Conference SV-NC95*, San Jose, CA, April 1995.
- [CSS94] Georg Carle, Jochen Schiller und Claudia Schmidt. Support for High-Performance Multipoint Multimedia Services. In *D. Hutchison, A. Danthine, H. Leopold, G. Colson (Eds.): Multimedia Transport and Teleservices; Lecture Notes in Computer Science 868*, Springer-Verlag, S. 219 - 240, International COST 237 Workshop, Wien, Österreich, November 1994.
- [CZ95] Georg Carle und Martina Zitterbart. ATM Adaptation Layer and Group Communication Servers for High-Performance Multipoint Services. In *7th IEEE Workshop on Local and Metropolitan Area Networks*, Marathon, Florida, U.S.A., März 1995.
- [FGM94a] Oliver Frick, Hans-Werner Gellersen und Max Mühlhäuser. Developing Cooperative Media-Integrated Software. In Bernd Wolfinger (Hrsg.), *Innovationen bei Rechen- und Kommunikationssystemen*, Hamburg, August 1994. 24. GI-Jahrestagung, Springer.

- [FGM94b] Oliver Frick, Hans-Werner Gellersen und Max Mühlhäuser. Multiuser and Multimodal Aspects of Multimedia. In Wolfgang Herzner und Frank Kappe (Hrsg.), *Multimedia/Hypermedia in Open Distributed Environments*. Eurographics Symposium, Graz, Springer, Juni 1994.
- [FR95] Oliver Frick und Martin Richartz. Exploiting High Speed Networks for Education. In *Workshop: Interactive and Distributed Multi-Media Systems on Highspeed Networks, on: Third International World-Wide Web Conference*, Darmstadt, april 1995.
- [FRG93] Oliver Frick, Tom Rüdebusch und Hans-W. Gellersen. Some Developers', End-Users', and General Issues in CSCW. In *ECSCW'93 Tools and Technologies Workshop*, Milano, Italy, September 1993.
- [Fri94] Oliver Frick. The Berkomp Multimedia Teleservices. Vortrag auf dem Eurographics Symposium, Graz, Juni 1994.
- [Gel93] Hans-W. Gellersen. Objekte in verteilten Systemen 9. Interner Bericht 24/93, Universität Karlsruhe, Fakultät für Informatik, 1993.
- [Gel94a] Hans-W. Gellersen. Aspects in User Interface Design for Mobile Multi-User Applications. In B. Wolfinger (Hrsg.), *Innovationen bei Rechen- und Kommunikationssystemen*, Informatik Aktuell, Seite 134-141. Springer-Verlag, Hamburg, August 28-September 2 1994. Tagungsband der 24. GI-Jahrestagung.
- [Gel94b] Hans-W. Gellersen. Integration of Concerns in User Interface Development for Mobile Multi-User Applications. In S. Howard und Y. Leung (Hrsg.), *Harmony through working together*, Melbourne, Australia, November 28-Dezember 1 1994. Seite 301-307. OZCHI'94 Conference Proceedings.
- [Gel94c] Hans-W. Gellersen. Multimodale User Interfaces und Telekooperation. Interner Bericht 20/94, Universität Karlsruhe, Fakultät für Informatik, 1994.
- [Gel95] Hans-W. Gellersen. Support of User Interface Design Aspects in a Framework for Distributed Cooperative Applications. In R. Taylor (Hrsg.), *Software Engineering and Human Computer Interaction*, Nr. 896 der Lecture Notes in Computer Science. Springer-Verlag, 1995.
- [Ger94] W. Gerteis. *Interaktive Führung beim Entwurf komplexer Software-Systeme*. Verlag Shaker. 1994.
- [GK94] S. Gessler und A. Kotulla. PDAs as mobile WWW-Browsers. In *2nd int. Conference on WWW*, Chicago (USA), October 1994. NCSA.
- [GKK95] A.G. Grosse, L. Keller und D.A. Kottmann. Verteilungstechnologische Grundlagen integrierter Systeme. In *Proc. SFB 346 Workshop Integrierte Planung flexibler Produktionssysteme*, 8. März 1995, Karlsruhe, 1995.
- [GM95a] Hans-W. Gellersen und M. Mühlhäuser. Design of Workplace-integrating User Interfaces based on Work Scenario Graphs. In D. Benyon und P. Palanque (Hrsg.), *Critical Issues in User Interface Systems Engineering*. Springer-Verlag, London, 1995.
- [GM95b] Hans-W. Gellersen und Max Mühlhäuser. Arbeitsplatzintegration und Medienintegration: Mensch-Computer-Interaktion in kooperativen Anwendungen. In Heinz-Dieter Böcker (Hrsg.), *Software-Ergonomie '95: Mensch-Computer-Interaktion, Anwendungsbereiche lernen voneinander*, Nr. 45 der Berichte des German Chapter of the ACM, Darmstadt, Februar20-23 1995. B.G. Teubner Stuttgart, Seite 143-162.
- [Kel93] L. Keller. Vom Name-Server zum Trader — Ein Überblick über Trading in verteilten Systemen. *Praxis der Informationsverarbeitung und Kommunikation (PIK)* 16(3), September 1993, Seite 122-133.
- [Kel94] L. Keller. Trading of Complex Services in Distributed Systems. In *5th IFIP/IEEE International Workshop on Distributed Systems: Operations & Management (DSOM'94)*, Toulouse, Frankreich, Oktober 1994.
- [KG94] Dietmar A. Kottmann und Stefan Gessler. Mobile Datenkommunikation und ihre Auswirkungen auf die Anwendungsunterstützung. *HMD — Theorie und Praxis der Wirtschaftsinformatik*, Heft 177, Mai 1994, Seite 84-97.
- [KG95] L. Keller und A.G. Grosse. Vermittlung zuverlässiger Dienste in Trading-basierten Systemen. In *Workshop Anwendungsunterstützung für heterogene Rechnernetze*, März 1995, Freiberg/Sachsen, 1995.
- [KK94a] L. Keller und D.A. Kottmann. Verwaltung lose-gekoppelter Datenbankcluster mittels Trading. In *Proc. 39. Internationales Wissenschaftliches Kolloquium*, Ilmenau, S. 194-199, 1994.
- [KK94b] D.A. Kottmann und L. Keller. Integrating Distribution and Mobility into an Object-Oriented Database. In Patel D., Sun Y., Patel S.: *Proc. International Conference on Object Oriented Information Systems (OOIS'94)*, Springer Verlag, London, Dezember, S. 350-363, 1994.
- [KK94c] D.A. Kottmann und L. Keller. Transparent Support for Object Mobility in an Object-Oriented Database. In *Proc. International Symposium on Applied Corporate Computing (ISACC'94)*, Monterrey, Mexiko, Oktober, S. 85-94, 1994.
- [KK95] L. Keller und D.A. Kottmann. Verteilung und Mobilität in einer objektorientierten Datenbank. In Franke K., Hübner U., Kalfa W.: *Proc. Kommunikation in verteilten Systemen (KIVS'95)*, Informatik Fachberichte, Springer Verlag, S. 285-299, 1995.
- [KKK⁺94] L. Keller, C. Kilger, D.A. Kottmann, G. Moerkotte, A.B. Schill, H.-D. Walter und A. Zachmann. Aktive und mobile Objekte als Modellierungskonzept für dezentrale Ingenieur Anwendungen. In 3.

- GI/ITG/GMA Fachtagung Softwaretechnik in Automation und Konstruktion — Datenbanken unter Realzeit- und Entwicklungsanforderungen (STAK'94), Illmenau, S. 75-100, März, 1994.*
- [KLW95] D.A. Kottmann, P.C. Lockemann und H.-D. Walter. Multi-Object Cooperation in Distributed Object Bases. Technischer Bericht, 16/95, Fakultät für Informatik, Universität Karlsruhe, 1995.
 - [Kot93] Dietmar A. Kottmann. Unterstützung der Datenhaltung in mobilen Systemen. In *Entwicklung und Management verteilter Anwendungssysteme*, Frankfurt am Main, Oktober 1993. GI/ITG Fachgruppe Kommunikation und verteilte Systeme.
 - [Kot95a] Dietmar A. Kottmann. Abgekoppelte Objekte in heterogenen Umgebungen mit mobilen Teilnehmern. In *Workshop Anwendungsunterstützung für heterogene Rechnernetze*, Freiberg/Sachsen, März 1995.
 - [Kot95b] Dietmar A. Kottmann. Datenmanagement im Mobile Computing. *HMD — Theorie und Praxis der Wirtschaftsinformatik*, Erscheint in Heft 184, Juli 1995.
 - [KS95] D.A. Kottmann und A.B. Schill. Flexible and Transparent Fault-Tolerance for Distributed Object-Oriented Applications. Technischer Bericht, 20/95, Fakultät für Informatik, Universität Karlsruhe, März 1995.
 - [KTZ94a] O. G. Koufopavlou, A. N. Tantawy und M. Zitterbart. A Comparison of Gigabit Router Architectures. Conference on High Performance Networking, Juni 1994.
 - [KTZ94b] O. G. Koufopavlou, A. N. Tantawy und M. Zitterbart. IP-Routing among Gigabit Networks. Second International Symposium on Interworking, Mai 1994.
 - [MGG95] Max Mühlhäuser, Hans-Werner Gellersen und Stefan Gessler. Distributed Client Architectures for advanced WWW applications. In *Interactive and Distributed Multi-Media Systems on Highspeed Network*, Darmstadt (Germany), April 1995.
 - [MM94] M. Mock und A. Müller. Generation of Distributed Services with Diogenes. In M. Medina und N. S. Borenstein (Hrsg.), *IFIP International Conference on Upper Layer Protocols, Architecture and Applications*, Seite 13 – 28. Spain, Juni 1994.
 - [Rö95] J. Röthig. *Kooperatives, adaptives Ressourcen-Management in ATM-Netzwerken*. VDI Verlag. 1995.
 - [SB93] Jochen Schiller und Torsten Braun. VLSI-Implementation Architecture for Parallel Transport Protocols. In *Workshop on VLSI in Communications*, Lake Tahoe, CA, September 1993. IEEE, Seite 76–77.
 - [Sch94] Jochen Schiller. VLSI-Implementierungsarchitektur für parallele Transportprotokolle. In *1. Arbeitstreffen Entwurf und Implementierung von Hochleistungs- Kommunikationssystemen*, Karlsruhe, Januar 1994. GI/ITG.
 - [Sei94a] J. Seitz. *Integration heterogener Netzwerkmanagementarchitekturen*. Nr. 289 der Fortschrittsberichte VDI, Reihe 10: Informatik/Kommunikationstechnik. VDI-Verlag, Düsseldorf. 1994.
 - [Sei94b] J. Seitz. *Netzwerkmanagement*. Nr. 2 der Thomsons's Aktuelle Tutorien (TAT). International Thomson Publishing, Bonn. Ausgabe 1, 1994.
 - [Sei94c] J. Seitz. OSF DCE und DME: Neue Industriestandards für Rechnernetze und verteilte Anwendungen — Teil 7: Eine Übersicht über das OSF-Distributed Management Environment. *unix/mail* 12(3), Juni 1994, Seite 179 – 184.
 - [Sei94d] J. Seitz. OSF DCE und DME: Neue Industriestandards für Rechnernetze und verteilte Anwendungen — Teil 8: Das „Traditional Management Framework“ von OSF/DME. *unix/mail* 12(4), August 1994, Seite 377 – 383.
 - [Sei94e] J. Seitz. OSF DCE und DME: Neue Industriestandards für Rechnernetze und verteilte Anwendungen — Teil 9: Das „Object Oriented Framework“ von OSF/DME. *unix/mail* 12(5), Oktober 1994, Seite 432 – 436.
 - [Sei94f] J. Seitz. OSF DCE und DME: Neue Industriestandards für Rechnernetze und verteilte Anwendungen — Teil 10: Managementanwendungen in OSF/DME. *unix/mail* 12(6), Dezember 1994, Seite 513 – 517.
 - [Sei94g] J. Seitz. Towards Integrated Network Management: A Generic Management Protocol. In *Fifth IFIP/IEEE International Workshop on Distributed Systems: Operations & Management DSOM'94*, 118, Route de Narbonne, 31062 Toulouse Cedex (France), 10. – 12. Oktober 1994. IFIP Working Group 6.6 Network Management; IEEE CNOM; Centre Interuniversitaire de Calcul de Toulouse; Conseil Régional Midi-Pyrénées; Hewlett Packard; Institut de Recherche en Informatique de Toulouse; Université Paul Sabatier, Université Paul Sabatier – IRIT/SIERA, Seite 6-1 — 6-17.
 - [Sei95] J. Seitz. OSF DCE und DME: Neue Industriestandards für Rechnernetze und verteilte Anwendungen — Teil 11: OSF/DME — Produkte und Zukunftsaussichten. *unix/mail* 13(1), Februar 1995, Seite 45 – 50.
 - [Sol94] G. Solvie. *Adaptive Kommunikationsunterstützung in offenen Systemen*. VDI Verlag. 1994.
 - [SSSZ94a] D. C. Schmidt, B. Stiller, T. Suda und M. Zitterbart. Configuring Function-based Communication Protocols for Multimedia Applications. IFIP Conference on Upper Layer Protocols, Architectures and Applications, Juni 1994.
 - [SSSZ94b] Douglas C. Schmidt, Burkhard Stiller, Tatsuya Suda und Martina Zitterbart. Configuring Function-based Communication Protocols for Multimedia Applications. In *IFIP 6.5 International Working Conference on Upper Layer Protocols, Architectures, and Applications, ULPAA'94*, Amsterdam, Netherlands, 1. – 3. June 1994. IFIP Transactions C-14, North Holland, Seite 361–367.

- [SSZ94] Claudia Schmidt, Burkhard Stiller und Martina Zitterbart. Towards Flexible Service-integrated Communication Subsystems. In *IASTED/ISMM International Conference on Distributed Multimedia Systems and Applications*, Honolulu, Hawaii, U.S.A., 15.-17. August 1994. Seite 361-367.
- [Sti94a] B. Stiller. CODCOM: A Code Configuration Manger in the Function-based Communication Subsystem. In *4th Open Workshop on High Speed Networks*, Brest, France, 7.-9. September 1994.
- [Sti94b] B. Stiller. *Flexible Protokollkonfiguration zur Unterstützung eines diensteintegrierenden Kommunikationssubsystems*, Band 10, no. 306. VDI, Düsseldorf, Germany. 16. February 1994.
- [Sti94c] B. Stiller. FuKSS: Ein funktionsbasiertes Kommunikationssystem zur flexiblen Konfiguration von Kommunikationsprotokollen. In *1. GI/ITG Arbeitstreffen zur Architektur und Implementierung von Hochleistungs-Kommunikationssystemen*, Karlsruhe, Germany, 17.-18. January 1994. Seite Session 3, Paper 1.
- [Sti94d] B. Stiller. PROCOM: A Protocol Configuration Manager in the Function-based Communication Subsystem. In *First International Workshop on High Performance Protocol Architectures*, Sophia-Antipolis, France, 15.-16. December 1994. Seite Session 3, Paper 9.
- [STZ94] D. N. Serpanos, A. N. Tantawy und M. Zitterbart. A High Performance Transparent Bridge. *ACM/IEEE Transactions in Networking* 2(4), August 1994, Seite 352 - 362.
- [SZ94] Claudia Schmidt und Martina Zitterbart. Towards Integrated QoS Management. In *First International Workshop on High Performance Protocol Architectures HIPPARCH '94*, Sophia Antipolis, Frankreich, Dezember 1994.
- [SZ95a] Jochen Schiller und Martina Zitterbart. Modular VLSI Implementation Architecture for High-Performance Communication Support. In *7th IEEE Workshop on Local and Metropolitan Area Networks*, Marathon, FL, März 1995. IEEE.
- [SZ95b] Claudia Schmidt und Martina Zitterbart. *Internetworking - Brücken, Router und Co.* Nr. 8 der Thomson Aktuelle Tutorien (TAT). International Thomson Publishing, Bonn. 1995.
- [SZ95c] Claudia Schmidt und Martina Zitterbart. QoS Management for Service Integrated Communication Systems. In *7th IEEE Workshop on Local and Metropolitan Area Networks*, Marathon, Florida, U.S.A., März 1995.
- [SZ95d] Claudia Schmidt und Martina Zitterbart. Reservierung von Netzwerk-Ressourcen - Ein Überblick über Protokolle und Mechanismen. *PIK - Praxis der Informationsverarbeitung und Kommunikation*, 1995.
- [TZ93a] A. N. Tantawy und M. Zitterbart. An Enhanced Transport Service Interface for Emerging Applications. Symposium on Global Networking, Dezember 1993.
- [TZ93b] A. N. Tantawy und M. Zitterbart. A Scheme for High Performance LAN Interconnection across MANs. *IEEE Journal on Selected Areas in Communications* 11(8), Oktober 1993, Seite 1133 - 1144.
- [WZ95] Hajo R. Wiltfang und Martina Zitterbart. IP-Prozessor für Gigabit-Router. In K. Franke, U. Hübner und W. Kalfa (Hrsg.), *Kommunikation in Verteilten Systemen*, TU Chemnitz-Zwickau, 22. - 24. Februar 1995. GI/ITG-Fachtagung, Springer-Verlag, Berlin Heidelberg, Seite 530-544.
- [Zei94] C. Zeidler. *Reflexive Konfigurationsverwaltung in einer objektorientierten verteilten Umgebung*. Shaker. 1994.
- [Zit93a] M. Zitterbart. Dienstintegrierende Hochleistungsnetze und -systeme für die Breitbandübertragung. Hochschul Computer Forum, Berlin, Oktober 1993.
- [Zit93b] M. Zitterbart. High Performance Transport Protocols and Services. Tutorial bei der 5th International Conference on Data Communication Systems and their Performance, Oktober 1993.
- [Zit93c] M. Zitterbart. Hochleistungskommunikation: Eine Herausforderung. *Praxis der Informationsverarbeitung und Kommunikation* 16(4), Dezember 1993, Seite 192.
- [Zit94a] M. Zitterbart. *High Performance Communications*, Kapitel Parallelism in Communication Subsystems, Seite 177 - 194. Kluwer Academic Publishers. 1994.
- [Zit94b] M. Zitterbart. *Hochleistungskommunikation; Band 1: Technologie und Netze*. Oldenbourg Verlag. 1994.
- [Zit94c] M. Zitterbart. Protokolle für die Hochleistungskommunikation. HMD-Heft 177: Hochgeschwindigkeitsnetze, 1994.
- [Zit94d] M. Zitterbart. Protokolle und Dienste für Hochleistungs-Kommunikationssysteme. 3. Workshop über wissenschaftliches Rechnen: Praxisorientierte Parallelverarbeitung, Oktober 1994.
- [Zit95] M. Zitterbart. *Flexible und effiziente Kommunikationssysteme für Hochleistungsnetze*. Thomson Publishing. 1995.